Réalisabilité Classique et protocoles réseaux

Philippe Hesse

UNIVERSITÉ PARIS DIDEROT - Paris 7
UFR de mathématiques

Thèse
pour l'obtention du diplôme de
Docteur de l'Université Paris Diderot

SPÉCIALITÉ :
MATHÉMATIQUES, LOGIQUE ET
FONDEMENTS DE L'INFORMATIQUE

RÉALISABILITÉ CLASSIQUE ET PROTOCOLES RÉSEAUX

présentée par

Philippe HESSE

soutenue le 17 juillet 2008 devant le jury composé de

M. Martin HYLAND	rapporteur
M. Jean-Louis KRIVINE	directeur de thèse
M. Yves LEGRANDGÉRARD	examinateur
M. Christophe RAFFALLI	examinateur
M. Laurent REGNIER	rapporteur

Remerciements

Je souhaite tout d'abord exprimer ma gratitude à celui qui a dirigé mon travail pendant ces quatres années, Jean-Louis Krivine. Cette thèse lui doit énormément. Je tiens à le remercier ici de m'avoir laissé une liberté totale dans mes travaux, et de m'avoir patiemment laissé commettre les erreurs nécessaires à l'appréhension d'un sujet aussi complexe. Ses enseignements ont été pour moi d'une grande richesse, et ont transformés ma vision des mathématiques.

Cette thèse doit également beaucoup à Yves Legrandgérard, qui lui a permis de trouver une cohérence entre les différentes thématiques qu'elle aborde. Je tiens également à le remercier d'avoir su me faire comprendre avec humour ce que de nombreux cours d'informatique n'avaient jamais su m'expliquer malgrès toute leur intelligence.

Merci à Laurent Regnier d'avoir insufflé ce travail il y a quatre ans, avec la force de sa simplicité et de son optimisme.

Merci à Martin Hyland et Laurent Regnier d'avoir accepté de rapporter cette thèse. Leur présence en ce jour est pour moi un grand honneur.

Merci à Christophe Raffalli d'avoir accepté de compléter ce jury. Sa présence est encore un grand honneur pour moi.

Merci à tous les membres du laboratoire PPS, des UFR de mathématiques et d'informatique de l'université Paris 7. Vous m'aurez tous appris que la réflexion est un sport collectif, qui se pratique dans la bonne humeur et avec générosité.

Merci également à tous ceux travaillant rue du Chevaleret que l'augmentation du prix du tabac n'a pas fait fléchir. Nos discussions quotidiennes furent pour moi source de joie et de remise en question.

Merci à mes amis, à ma famille, à tous ceux que j'aime. Votre présence et nos échanges sauront toujours réchauffer mon cœur et animer ma carcasse. J'espère qu'en chacun des jours qui nous seront donnés, mon regard pourra vous rendre ce que mes mots ne savent exprimer.

Merci enfin à vous d'avoir ouvert cette thèse. Bien que vous en lisiez les meilleurs mots, j'espère que vous prendrez le temps de les dépasser, et que vous saurez me pardonner les coquilles que vous ne manquerez pas de relever.

Table des matières

Chapitre 1

Introduction

Un bref historique

L'isomorphisme de Curry-Howard, ou *correspondance preuves/programmes*, établit une correspondance entre la logique formelle et le calcul symbolique. Il se fonde sur une interprétation fonctionnelle de l'implication : l'énoncé $A \implies B$ peut être associé à l'ensemble des fonctions de domaine A et d'image B. En effet, déduire B des propositions A et $A \implies B$ peut être vu comme l'application d'une telle fonction à un élément de A, et prouver $A \implies B$ en déduisant B de A revient à construire un élément de B en partant d'un élément de A. Il fut d'abord formalisé entre d'une part la *logique intuitionniste propositionnelle* munie du seul connecteur $\to$, et d'autre part le *λ-calcul* d'Alonzo Church.

Dans ce calcul propositionnel, les règles de construction des preuves sont les suivantes :

$$\frac{}{\Gamma, A \vdash A} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B} \qquad \frac{\Gamma \vdash A \to B \qquad \Gamma \vdash A}{\Gamma \vdash B}$$

Le langage servant de support à son interprétation est le λ-calcul, dont la syntaxe est donnée par la grammaire suivante :

$$u, v, w := x | (u)v | \lambda x\ w$$

où u, v et w représentent des *termes*, x une *variable*. $(u)v$ est alors interprété comme l'application d'une fonction u à un argument v, et $\lambda x\ w$ comme la construction d'une fonction qui à x associe w. On bâtit un modèle de calcul sur cette syntaxe en définissant la règle d'*évaluation* suivante, appelée *β-réduction* :

$$(\lambda x\ w)v \succ w[v/x]$$

Appliquer une fonction de x à un terme v revient donc à remplacer x par v dans l'expression définissant cette fonction. Mais la structure du λ-calcul permet de représenter bien plus que des fonctions au sens intuitif du terme, puisque par exemple le terme $\lambda x(x)x$ est syntaxiquement correct. Pour restreindre cette représentation aux fonctions, il suffit d'associer un domaine et

une image aux termes lors de leur construction, et de n'autoriser que les règles de construction suivantes :

$$\frac{}{\Gamma, x : A \vdash x : A} \qquad \frac{\Gamma, x : A \vdash w : B}{\Gamma \vdash \lambda x\, w : A \to B} \qquad \frac{\Gamma \vdash u : A \to B \qquad \Gamma \vdash v : A}{\Gamma \vdash (u)v : B}$$

On obtient à nouveau les règles de dérivation du calcul propositionnel intuitionniste, « décorées » avec des termes qui donnent un historique de la preuve en cours. La formule associée à un terme est appelée son *type*, et le terme lui-même est dit *typable*.

Haskell Curry, puis William Howard montrèrent que l'élimination des coupures d'une preuve correspond à la β-réduction du terme associé, ce qui assure la terminaison de l'évaluation d'un terme typable et la conservation du type au cours de sa réduction. La cohérence très satisfaisante de ce modèle constitue l'acte de naissance de la correspondance preuves/programmes.

Un programme est un objet formel, qui a pour but d'être exécuté par une machine symbolique construite sur une machine concrète. L'isomorphisme de Curry-Howard est une manière d'inscrire la programmation dans un cadre mathématique, et ainsi de prouver formellement la *correction des programmes* (représentés par des termes), c'est-à-dire leur adéquation à une *spécification* (ou type). Pris au pied de la lettre, il propose d'utiliser les mathématiques elles-mêmes comme espace de représentation des programmes et de leur exécution. Mais la logique propositionnelle intuitionniste est encore un canevas trop rudimentaire pour broder une telle œuvre.

Les travaux de Jean-Yves Girard ont alors permis d'étendre le système de types à la logique intuitionniste d'ordre deux, en conservant comme modèle de calcul les λ-termes munis de la β-réduction. Dans ce cadre les termes typés, et plus généralement les termes *réalisant* un type donné, mènent à des calculs exacts sur les *types de données*. Apparaît également la notion de *polymorphisme*, duale de la quantification universelle. Viendra ensuite la création de la *logique linéaire*, logique intuitionniste où les coupures et leur élimination sont finement décomposées, qui offrit d'autres perspectives à l'isomorphisme de Curry-Howard.

C'est ensuite grâce à Timothy Griffin que la correspondance fut étendue à la logique classique, avec la découverte de la coïncidence entre le tiers-exclu et les opérateurs de contrôle des langages impératifs usuels. Il fallut alors enrichir le modèle et les règles de calcul afin de pouvoir typer correctement la loi de Pierce. On peut notamment citer le *$\lambda\mu$-calcul* de Michel Parigot, dans lequel la correction du calcul est encore vérifiée.

Mais les démonstrations mathématiques ne se construisent pas seulement

avec des règles de démonstration, encore faut-il se donner des axiomes, et donc des *instructions* associées pour pouvoir enfin mettre en œuvre la correspondance. La *réalisabilité classique*, introduite par Jean-Louis Krivine, permet d'associer un programme à chaque démonstration faite à partir de n'importe quel système axiomatique, en ce sens que les axiomes de ZFC ont pu être réalisés dans ce cadre.

Le langage de programmation est alors un λ-calcul enrichi de différentes instructions, chacune représentant le contenu opérationnel d'un axiome (par exemple, un opérateur de contrôle pour la loi de Pierce). L'exécution n'est plus compatible avec la β-réduction, et ne décrit donc plus l'élimination des coupures. Elle est modélisée par une machine symbolique qui exécute les programmes, en assurant toujours la correction du calcul sur les types de données. Ce cadre correspond de très près à la programmation impérative usuelle, et l'on voit apparaître les notions fondamentales de celle-ci au fur et à mesure que la théorie se développe : opérateurs de mise en mémoire, types de données, pointeurs, signature d'un fichier, horloge système, utilisation d'une mémoire globale...

A chaque théorème correspond désormais un comportement que partagent tous les programmes qui le réalisent. Il reste évidemment à pouvoir exprimer et interpréter celui-ci ; c'est ce que l'on appelle le *problème de la spécification*. De même, il reste à tracer des correspondances internes à celle entre les preuves et les programmes : à quels domaines des mathématiques correspondent les différentes thématiques de l'informatique ?

Mais la réalisabilité classique ne permet pas seulement d'étudier l'isomorphisme de Curry-Howard : c'est également un outil logique puissant, qui propose de comprendre le raisonnement mathématique par l'étude des mécanismes qu'il induit. Il s'agit d'une extension du *forcing* (plus précisément un cas dégénéré d'une extension de celui-ci), dans laquelle les conditions sont des ensembles de termes, et qui permet de construire des modèles de ZFC.

Signalons enfin que l'isomorphisme de Curry-Howard établit plus qu'une correspondance entre les preuves et les programmes : un troisième domaine tend à se joindre à eux. Il s'agit de la notion de *jeu*, qui est par exemple utilisée en sémantique pour modéliser les interactions entre un programme et son environnement, mais qui correspond également à la notion de preuve en calcul des séquents. Ce domaine possède de nombreuses interprétations à la lumière de la correspondance, et notamment dans le cadre de la réalisabilité classique.

Plan de la thèse

Cette thèse étudie différents aspects de la réalisabilité classique, suivant trois axes principaux :

- l'illustration des possibilités descriptives de la théorie, avec l'étude de nombreuses spécifications ;
- l'interprétation des jeux en réalisabilité, à travers leur application à la spécification des protocoles de la couche transport des réseaux ;
- l'axiome du choix dépendant, dont on réalise une expression sur les individus d'un modèle, qui est ensuite employée pour extraire un terme d'une preuve formelle d'un cas particulier du théorème de Herbrand.

Le chapitre 2 présente le principe de la réalisabilité classique dû à Krivine, développé ici à partir d'un modèle de l'arithmétique du second ordre. On illustre ensuite les techniques usuelles de raisonnement dans ce formalisme, à travers le rappel d'un certain nombre de résultats fondamentaux. On présente notamment la notion de *modèle de la réalisabilité*, et l'on prouve que ceux-ci sont bien fondés mais ne satisfont pas l'axiome de récurrence. On montre enfin que la réalisabilité mène à des calculs exacts sur le type *entier*, en rappelant les spécifications de certaines formules.

Le chapitre 3 illustre les possibilités descriptives offertes par la théorie en termes de spécification. Considérant que les modèles de la réalisabilité ne satisfont pas l'axiome de récurrence, on y définit par bisimilarité une relation permettant d'obtenir de tels modèles par passage au quotient. Cette relation correspond à un prédicat qui peut être défini par point fixe, de sorte que les valeurs de vérité associées soient particulièrement simples. On réalise ensuite quelques formules exprimant des propriétés du quotient (transitivité de l'ordre, commutativité de l'addition), et l'on étudie précisément les spécifications associées.

Le chapitre 4 présente une notion de jeu adaptée au formalisme de la réalisabilité, et en fait inspirée de celle-ci comme le montre le chapitre suivant. A chaque formule du premier ordre écrite dans un ω-langage est associé un jeu entre deux joueurs, désignés par les symboles $\forall$ et $\exists$, de sorte que l'ω-validité de la formule équivaut à l'existence d'une stratégie gagnante pour $\exists$. A travers l'étude de quelques exemples, on montre ensuite que ces jeux permettent de spécifier de manière claire et précise les principes fondamentaux des protocoles de la couche transport des réseaux. On donne les caractéristiques de cette modélisation, que l'on illustre en considèrant le problème de l'envoi d'un ou de multiples paquets selon le principe de l'acquittement, ainsi que celui de l'initialisation d'une session (*three-way handshake*).

Le chapitre 5 présente une formalisation des jeux du chapitre 4 dans le cadre de la réalisabilité. On montre comment résoudre grâce à eux le problème de la spécification dans le cas des formules valides : chaque terme réalisant une telle formule implémente une partie dans le jeu associé, à l'aide d'instructions interactives, pour laquelle la joueuse $\exists$ possède une stratégie gagnante. On étudie ensuite certains processus associés aux exemples traités dans le chapitre précédent.

Dans le chapitre 6, on réalise une formule exprimant l'axiome du choix sur les individus d'un modèle. On adapte pour cela la technique développée pour réaliser cet axiome au second ordre. On introduit alors une nouvelle instruction, pouvant être interprétée comme un programme utilisant un algorithme de signature, et qui correspond au contenu opérationnel de l'axiome.

Le chapitre 7 illustre comment transformer une preuve formelle complexe en un programme. On y considère une formule Π^1_2 qui est un cas particulier du théorème de Herbrand, dont la démonstration nécessite l'axiome du choix au premier ordre, et donc l'utilisation de l'instruction associée. On verra que le terme obtenu effectue une opération très générale qui peut être interprétée dans le cadre des protocoles réseaux.

Chapitre 2

Réalisabilité classique

Nous commençons par rappeler les principes de la réalisabilité classique, tels qu'ils sont donnés dans [24]. Nous illustrerons ensuite les méthodes usuelles de raisonnement dans ce formalisme, à travers quelques résultats fondamentaux.

2.1 Définitions

2.1.1 Termes et piles

Nous allons dans cette partie définir le λ-calcul qui sera utilisé dans la suite. On se donne pour cela un nombre dénombrable de λ-variables notées x, y, z,... ainsi qu'un ensemble noté ι de constantes supplémentaires appelées *instructions* dont l'une est notée cc, en référence à la primitive call/cc (pour *call with current continuation*) du langage SCHEME. Les termes de ce calcul qui seront utilisés par la suite ne seront que des termes *clos*.

Définition 2.1.1.
On appelle QP l'ensemble des termes clos du λ-calcul construits récursivement à partir des λ-variables et de l'ensemble d'instructions ι, c'est-à-dire les termes clos donnés par la grammaire suivante :

$$t, u := \ x \mid \lambda x\ t \mid (t)u \mid i \quad \textit{(pour } i \in \iota\textit{)}$$

Les éléments de QP seront appelés des quasi-preuves.

Notations : On dénotera par $(t)u_1 \dots u_n$ le terme $(\dots((t)u_1)u_2 \dots u_n)$, c'est-à-dire l'application de t à n arguments $u_1, \dots, u_n$.
On désignera encore par I le λ-terme $\lambda x\ x$.

On se donne alors un nouvel ensemble de constantes noté Π_0 disjoint de ι, dont les éléments seront appelés des *constantes de pile*, ainsi qu'une nouvelle constante notée k (qui n'est pas considérée comme une instruction).

Définition 2.1.2.
On appelle L l'ensemble des termes clos du λ-calcul construits à partir de l'ensemble de constantes $\iota \cup \Pi_0 \cup \{\mathsf{k}\}$, c'est-à-dire l'ensemble des termes clos donnés par la grammaire suivante :

$$t, u := \ x \mid \lambda x\ t \mid (t)u \mid i \quad \textit{(pour } i \in \iota \cup \Pi_0 \cup \{\mathsf{k}\}\textit{)}$$

Définition 2.1.3.
On appelle continuation un terme ***clos*** *de L de la forme $(\mathsf{k})t_1 \dots t_n\pi_0$ avec $n \in \mathbb{N}$, $t_1, \dots, t_n \in L$ et $\pi_0 \in \Pi_0$.*

Définition 2.1.4.
On appelle λ_c-terme tout terme clos τ de L ayant les propriétés suivantes :
- *chaque occurrence de la constante* k *dans τ est le début d'un sous-terme de τ qui est une continuation.*
- *chaque occurrence d'une constante de pile dans τ est la fin d'un sous-terme de τ qui est une continuation.*

On désigne par Λ_c l'ensemble des λ_c-termes.

Les λ_c-termes sont les termes qui seront utilisés par la suite ; on se référera à eux simplement en tant que « termes ». Les quasi-preuves sont donc les λ_c-termes qui ne contiennent pas le symbole k ou, ce qui revient au même, ne contiennent pas de constante de pile.

Lemme 2.1.5.
On a les assertions suivantes :

i) $QP \subset \Lambda_c$.

ii) *Si* $(t)u \in \Lambda_c$ *et* $u \notin \Pi_0$, *alors* $t \in \Lambda_c$ *et* $u \in \Lambda_c$.

iii) *Si* $(t)u \in \Lambda_c$ *et* $u \in \Pi_0$, *alors* $(t)u$ *est une continuation.*

iv) *Si* u, $\lambda x\ t \in \Lambda_c$, *alors* $t[u/x] \in \Lambda_c$.

v) *Si* $\pi_0 \in \Pi_0$ *et* $t_1, \dots, t_n \in \Lambda_c$, *alors* $(\mathsf{k})t_1 \dots t_n\pi_0 \in \Lambda_c$. *Réciproquement, si* $(\mathsf{k})t_1 \dots t_n\pi_0 \in \Lambda_c$ *et* $\pi_0 \in \Pi_0$, *alors* $t_1, \dots, t_n \in \Lambda_c$.

Démonstration. *i)* Trivial.

ii) Considérons une occurrence de k dans t (resp. u) ; cette occurrence est donc dans $(t)u$. Comme $(t)u$ est dans Λ_c, c'est le début d'une continuation $((\mathsf{k})t_1 \dots t_n)\pi_0$ qui est un sous-terme de $(t)u$. Comme u ne peut être π_0, cette continuation ne peut être le terme $(t)u$ lui-même ; il s'agit donc d'un sous-terme de t (resp. u). On raisonnerait exactement de même en partant d'une constante de pile π_0 apparaissant dans t ou u.

iii) On a $u = \pi_0 \in \Pi_0$, et cette occurrence est à la fin d'un sous-terme de $(t)\pi_0$ qui est une continuation : celle-ci ne peut que être le terme lui-même.

iv) On considère une occurrence de k dans $t[u/x]$; elle correspond donc à une occurrence de k qui est soit dans $\lambda x\ t$, soit dans u. Mais ces deux termes sont dans Λ_c, cette occurrence est donc le début d'une continuation $((\mathsf{k})t_1 \dots t_n)\pi_0$ qui est un sous-terme soit de $\lambda x\ t$ soit de u. S'il s'agit

d'un sous-terme de u, elle est donc aussi un sous-terme de $t[u/x]$. Si elle est un sous-terme de $\lambda x\ t$, alors l'occurrence considérée de k dans $t[u/x]$ est le début de la continuation $((\mathsf{k})t_1[u/x]\dots t_n[u/x])\pi_0$, qui est un sous-terme de $t[u/x]$. On raisonnerait de même en considérant l'occurrence d'une constante de pile.

v) On considère une occurrence de k dans $((\mathsf{k})t_1\dots t_n)\pi_0$. S'il s'agit de l'occurrence que l'on voit apparaître ici il n'y a rien à démontrer. Sinon, il s'agit d'une occurrence apparaissant dans l'un des t_i. Mais ceux-ci étant des λ_c-termes, il s'agit du début d'une continuation qui est un sous-terme de t_i, et qui est donc également un sous-terme de $((\mathsf{k})t_1\dots t_n)\pi_0$. Réciproquement, considérons une occurrence de k dans l'un des t_i. Il s'agit donc du début d'une continuation qui est un sous-terme de $((\mathsf{k})t_1\dots t_n)\pi_0$, puisque ce terme est dans Λ_c. Mais cette continuation ne peut alors qu'être un sous-terme de t_i. On raisonne encore une fois de même si l'on considère une occurrence d'une constante de pile.

□

Définition 2.1.6.
On note Π *l'ensemble construit récursivement par les deux règles suivantes :*
- $\Pi_0 \subset \Pi$.
- *si* $\pi \in \Pi$ *et* $t \in \Lambda_c$, *alors* $(t,\pi) \in \Pi$.

On appele « pile » tout élément de Π. *Les éléments de* Π_0 *seront eux appelés des « piles vides ».*

Si $\pi \in \Pi$ et $t \in \Lambda_c$, la pile (t,π) sera notée $t \cdot \pi$. Toute pile est donc de la forme $t_1 \cdot \dots \cdot t_n \cdot \pi_0$.

Notation : Le lemme 2.1.5 permet d'associer à chaque continuation $(\mathsf{k})t_1\dots t_n\pi_0$ qui est un λ_c-terme (ou, ce qui revient au même, apparaît dans un λ_c-terme), une *pile* π (en l'occurrence $t_1 \cdot \dots \cdot t_n \cdot \pi_0$). On dénotera donc dans la suite le terme $(\mathsf{k})t_1\dots t_n\pi_0$ par k_π.

Lemme 2.1.7.
Tout terme de Λ_c *est de l'une des formes suivantes, ces cas s'excluant l'un l'autre :*
i) $(t)u$ *avec* $t,u \in \Lambda_c$
ii) k_π *avec* $\pi \in \Pi$, *c'est-à-dire qu'il s'agit d'une continuation,*
iii) $\lambda x\ t$,
iv) α, *qui est une instruction (par exemple* cc*).*

Démonstration. Découle aisément de la définition de Λ_c et du lemme 2.1.5. □

2.1.2 Processus et machine symbolique

On n'utilise pas ici la β-réduction, mais une machine symbolique qui permet d'exécuter un *terme* muni d'une *pile*, celle-ci correspondant au contexte d'évaluation. L'exécution dans cette machine d'un λ-terme usuel muni d'une pile vide correspond à sa réduction de tête faible. Pour plus de détails, on se reportera à [26].

Définition 2.1.8.
On appelle processus tout élément de $\Lambda_c \times \Pi$. Un processus (τ, π) sera noté $\tau \star \pi$. τ est appelé la « tête » du processus, et π la « pile courante ». On notera également $\Lambda_c \star \Pi$ l'ensemble des processus.

On peut désormais définir une règle d'exécution des processus, qui sera notée $\succ$. On définit déjà la règle définissant « un pas d'exécution. »

Définition 2.1.9.
On note $\dot{\succ}$ la plus petite relation binaire sur $\Lambda_c \star \Pi$ telle que :
i) $(t)u \star \pi \dot{\succ} t \star u \cdot \pi$ ***(empilement)*** *;*
ii) $\lambda x\ t \star u \cdot \pi \dot{\succ} t[u/x] \star \pi$ ***(désempilement)*** *;*
iii) $\mathsf{cc} \star t \cdot \pi \dot{\succ} t \star \mathsf{k}_\pi \cdot \pi$ ***(sauvegarde de la pile)*** *;*
iv) $\mathsf{k}_\pi \star t \cdot \rho \dot{\succ} t \star \pi$ ***(restauration de la pile)***.
Pour les autres instructions, la règle sera donnée au cas par cas lorsque celles-ci seront introduites.

Remarques :
- Le lemme 2.1.5 permet d'assurer que cette relation est bien définie sur $\Lambda_c \times \Pi$.
- Le lemme 2.1.7 permet lui de prouver que pour chaque processus p, il existe au plus un processus p' tel que $p \dot{\succ} p'$.
- L'instruction cc permet de mémoriser la pile courante en créant le terme k_π ; celui-ci permet la réinstallation de cette pile lorsqu'il arrive en tête. Un programme peut donc manipuler explicitement son flot de contrôle à l'aide cette instruction.

Définition 2.1.10.
On note $\succ$ la clôture réflexive et transitive de la relation $\dot{\succ}$. On se référera à la relation $p \succ p'$ en écrivant « p se réduit en p' », ou « l'exécution de p mène à p' ».

2.1.3 Types et règles de typage

Notations : On considère un ensemble $\mathscr{N}$ quelconque.
Les éléments de $\mathscr{N}$ seront notés n, p,...
On se donne un symbole de fonction f pour chaque application de $\mathscr{N}^k$ dans $\mathscr{N}$, pour $k \geqslant 0$. Les symboles de fonctions 0-aire seront appelés des *symboles de constantes.* Les éléments de $\mathscr{P}(\Pi)^{\mathscr{N}^k}$ seront eux notés R, Φ, Ψ,... et seront appelées des *prédicats.*
Les formules seront notées F, G,...
Les variables du premier ordre (appelées aussi *variables d'individus*) utilisées dans les formules seront notées x, y,... et les variables du second ordre (appelées aussi *variables de prédicat*) X, Y...
Les variables du second ordre d'arité 0 seront appelées *variables propositionnelles.*

Définition 2.1.11.
On appelle $T_{\mathscr{N}}$ l'ensemble des termes définis récursivement de la manière suivante :
- *une variable d'individus est un terme ;*
- *un symbole de constante est un terme ;*
- *si f désigne un symbole de fonction k-aire pour $k \geqslant 1$, et si $\tau_1, \ldots, \tau_k$ désignent des termes, alors $f(\tau_1, \ldots, \tau_k)$ est un terme.*

Remarque : On prend la même notation pour une fonction $f : \mathscr{N}^k \to \mathscr{N}$ et pour le symbole la représentant dans $T_{\mathscr{N}}$. On prendra donc garde à ne pas confondre un « terme » avec sa « valeur », cette dernière étant un élément de $\mathscr{N}$.

Définition 2.1.12.
On appelle types ou formules à paramètres les objets construits récursivement de la manière suivante :

1. *Si $R \in \mathscr{P}(\Pi)^{\mathscr{N}^k}$ et si $\tau_1, \ldots, \tau_k$ sont des termes, alors $R(\tau_1, \ldots, \tau_k)$ est une formule.*
2. *Si X est une variable de prédicat k-aire et si $\tau_1, \ldots, \tau_k$ sont des termes, alors $X(\tau_1, \ldots, \tau_k)$ est une formule.*
3. *Si G et F sont des formules, alors $G \to F$ est une formule.*
4. *Si $\Lambda \subseteq \Lambda_c$ et si F est une formule, alors $\Lambda \to F$ est une formule.*
5. *Si x est une variable d'individus et si F est une formule, alors $\forall x F$ est une formule.*
6. *Si X est une variable de prédicat et si F est une formule, alors $\forall X F$ est une formule.*

Les types considérés sont donc les formules de la logique classique du second ordre écrites avec les seuls symboles $\forall$ et $\to$, à ceux-ci près que l'on s'autorise à placer des ensembles de termes *à gauche* des flèches. Cet ajout est des

plus naturels par rapport à la notion de valeur de vérité introduite en réalisabilité classique, et permettra par la suite de simplifier les termes issus d'une preuve.
Le fait de considérer les seuls connecteurs $\forall$ et $\rightarrow$ est intimement lié à la notion de valeur de vérité qui sera définie (Cf. la définition 2.1.19), et permet encore de simplifier les termes obtenus.

Définition 2.1.13.
Une formule construite sans la règle 3 *sera appelée une formule usuelle.*

Notations : La formule $F_0 \rightarrow (F_1 \rightarrow (... \rightarrow (F_n \rightarrow G)...))$ est notée $F_0, F_1, ..., F_n \rightarrow G$.
$\bot$ (lire « faux ») est défini par la formule $\forall X\ X$;
$\neg F$ par $F \rightarrow \bot$;
$F \vee G$ par $\forall X[(F \rightarrow X), (G \rightarrow X) \rightarrow X]$;
$F \wedge G$ par $\forall X[(F, G \rightarrow X) \rightarrow X]$;
$\exists y F(y)$ par $\forall X[\forall y(F(y) \rightarrow X) \rightarrow X]$ et
$\exists Y F(Y)$ par $\forall X[\forall Y(F(Y) \rightarrow X) \rightarrow X]$.
Enfin, l'expression $\exists y(F_1(y), \ldots, F_n(y))$ dénote la formule
$\forall X[\forall y(F_1(y), \ldots, F_n(y) \rightarrow X) \rightarrow X]$.
Dans toutes ces formules, X désigne une variable propositionnelle et Y une variable de prédicat d'arité quelconque.
En outre, l'égalité $x = y$ est définie par la formule de Leibniz

$$\forall X(X(x) \rightarrow X(y))$$

où X est une variable de prédicat d'arité 1.

Définition 2.1.14.
Soient $x_1, \ldots, x_k$ des variables d'individu, X une variable de prédicat d'arité k, G et $F(x_1, \ldots, x_k)$ des formules quelconques, les variables libres (du premier ordre) de F étant parmi $x_1, \ldots, x_k$. On définit la formule $G[F/X(x_1, \ldots, x_k)]$ par récurrence de la manière suivante :

- *Si X n'est pas libre dans G, alors $G[F/X(x_1, \ldots, x_k)]$ est G.*
- *Si G est $X[t_1, \ldots, t_k]$, alors $G[F/X(x_1, \ldots, x_k)]$ est $F[t_1/x_1, \ldots, t_k/x_k]$.*
- *Si G est de la forme $A \rightarrow B$, alors $G[F/X(x_1, \ldots, x_k)]$ est $A[F/X(x_1, \ldots, x_k)] \rightarrow B[F/X(x_1, \ldots, x_k)]$. Dans le cas particulier où A désigne un ensemble de termes, $A[F/X(x_1, \ldots, x_k)]$ est par définition A.*
- *Si G est de la forme $\forall y A$, alors $G[F/X(x_1, \ldots, x_k)]$ est $\forall y\ A[F/X(x_1, \ldots, x_k)]$, où l'on a supposé, quitte à renommer la variables liée y, que celle-ci n'était pas parmi $x_1, \ldots, x_k$.*
- *Enfin si G est de la forme $\forall Y A$, alors $G[F/X(x_1, \ldots, x_k)]$ est $\forall Y\ A[F/X(x_1, \ldots, x_k)]$, où Y est une variable de prédicat qui n'est pas X, et qui est supposée ne pas être libre dans F.*

Notation : Si R est une variable de prédicat d'arité k ou un élément de $\mathscr{P}(\Pi)^{\mathscr{N}^k}$, on écrira $A[R/X]$ au lieu de $A[R/X(x_1, \ldots, x_k)]$.

On donne enfin les règles typage des λ_c-termes associées aux règle de déduction en logique classique du second ordre.

Définition 2.1.15.
On appelle contexte une expression de la forme $t_1 : A_1, \ldots, t_n : A_n$, où chacun des t_i est un élément de L, et chacun des A_i est une formule écrite sans ensemble de termes.

Définition 2.1.16.
On appelle relation de typage la plus petite relation sur les triplets de la forme (Γ, t, A), que l'on notera $\Gamma \vdash t : A$, où Γ est un contexte, t un élément de L et A une formule écrite sans ensemble de termes, satisfaisant les règles d'inférence suivantes :

1. $t_1 : A_1, \ldots, t_n : A_n \vdash t_i : A_i \ (1 \leqslant i \leqslant n)$;
2. $\Gamma \vdash t : A \to B, \Gamma \vdash u : A \implies \Gamma \vdash (t)u : B$;
3. $\Gamma, x : A \vdash t : B \implies \Gamma \vdash \lambda x\, t : A \to B$;
4. $\Gamma \vdash t : (A \to B) \to A \implies \Gamma \vdash (\mathsf{cc})t : A$;
5. $\Gamma \vdash t : A \implies \Gamma \vdash t : \forall x A$, *si x n'est libre dans aucune des formules apparaissant dans* Γ ;
6. $\Gamma \vdash t : A \implies \Gamma \vdash t : \forall X A$, *si X n'est libre dans aucune des formules apparaissant dans* Γ ;
7. $\Gamma \vdash t : \forall x A \implies \Gamma \vdash t : A[\tau/x]$, *pour tout terme τ de $T_{\mathscr{N}}$* ;
8. $\Gamma \vdash t : \forall X A \implies \Gamma \vdash t : A[F/X(x_1, \ldots, x_k)]$, *pour toute formule F.*

La règle 4 se fonde sur l'interprétation de Griffin des opérateurs de contrôle, alors que la règle 8 repose sur l'interprétation de Takeuti du schéma de compréhension.

On appelle *séquent*, toute expression de la forme $\Gamma \vdash t : F$, où Γ est un contexte, t un élément de L et F une formule.

Remarque : Les seuls termes clos qui sont typables sont évidemment les quasi-preuves, puisque qu'on ne donne pas de type aux continuations.

2.1.4 Réalisabilité classique

On considère toujours un ensemble $\mathscr{N}$ fixé.

Définition 2.1.17.
Un ensemble $\perp\!\!\!\perp$ (lire « bottom ») de processus est dit « saturé » s'il est clos par antéréduction pour les quatre règles d'exécution de la définition 2.1.9 ; c'est-à-dire que considérant une exécution $p' \succ p$ n'utilisant que ces règles, on a :

$$p \in \perp\!\!\!\perp \implies p' \in \perp\!\!\!\perp$$

Définition 2.1.18.
On considère une instruction κ. Un ensemble $\perp\!\!\!\perp$ de processus est dit « saturé pour κ » s'il est saturé, et de plus clos par antéréduction pour la règle d'exécution de κ.

On considérera par la suite de nombreuses instructions, mais on ne demandera à l'ensemble $\perp\!\!\!\perp$ d'être saturé pour leurs règles d'exécutions que si elles correspondent à l'ajout d'un nouvel axiome, c'est-à-dire si elles seront utilisées pour écrire des quasi-preuves. Dans le chapitre 6 par exemple, on introduira de nouvelles instructions qui permettront de réaliser un axiome du choix au premier ordre, mais uniquement pour les ensembles $\perp\!\!\!\perp$ saturés pour celles-ci.

On commence par fixer arbitrairement un ensemble saturé de processus, que l'on notera $\perp\!\!\!\perp$, qui est le paramètre dont dépend la définition de la notion de réalisabilité.
Nous allons introduire une notion de valeur de vérité pour les formules, définie par récurrence simultanément à la définition de la relation de réalisabilité.

Définition 2.1.19.
Soit $\perp\!\!\!\perp$ un ensemble saturé.
Soit F une formule close, éventuellement avec des paramètres du premier (c'est-à-dire des éléments de $\mathcal{N}$) et du second ordre (c'est-à-dire des éléments de $\mathcal{P}(\Pi)^{\mathcal{N}^k}$).
On définit ci-dessous, par induction sur F, sa valeur de vérité que l'on notera $||F||$ et qui est un sous ensemble de Π.
On définit simultanément l'ensemble des λ_c-termes qui réalisent F, qui sera noté $|F|$. Celui-ci est défini à chaque étape de l'induction, une fois que la valeur de vérité d'une formule G est définie, par $|G| = \{t \in \Lambda_c; \forall \pi \in ||G||, t \star \pi \in \perp\!\!\!\perp\}$.
Les étapes de l'induction sont :

- *Si $\tau_1, \dots \tau_k$ sont des termes clos dans $T_{\mathcal{N}}$, et si $R \in \mathcal{P}(\Pi)^{\mathcal{N}^k}$, alors on pose $||R(\tau_1, \dots, \tau_k)|| = R(n_1, \dots, n_k) \subset \Pi$, où n_i désigne la « valeur » du terme τ_i dans $\mathcal{N}$ pour $1 \leqslant i \leqslant k$;*
- *$||A \to B||$ est défini comme étant $\{t \cdot \pi \ ; \ t \in |A|, \ \pi \in ||B||\}$;*
- *Si Λ désigne un sous ensemble de Λ_c, $||\Lambda \to B||$ est défini comme étant $\{t \cdot \pi; \ t \in \Lambda, \ \pi \in ||B||\}$;*
- *$||\forall x A||$ est défini comme étant $\bigcup_{n \in \mathcal{N}} ||A[n/x]||$;*
- *$||\forall X A||$ est défini comme étant $\bigcup_{R \in \mathcal{P}(\Pi)^{\mathcal{N}^k}} ||A[R/X]||$, si X désigne une variable de prédicat d'arité k.*

Remarque : La valeur de vérité d'une formule peut être vue comme l'ensemble des piles qui s'opposent à (la véracité de) celle-ci. Un terme réalisera alors cette formule s'il s'oppose lui-même à cet ensemble de piles.

Par la suite, dans le but de simplifier les notations, on laissera la notion de $\perp\!\!\!\perp$ implicite lorsque celui-ci sera fixé et quelconque.

Lemme 2.1.20.
On a les résultats suivants :

i) $|\forall x A| = \bigcap_{n\in\mathcal{N}} |A[n/x]|$;

ii) $|\forall X A| = \bigcap_{R\in\mathscr{P}(\Pi)^{\mathcal{N}^k}} |A[R/X]|$;

iii) Quelles que soient les formules closes F et G, $||F|| \subseteq ||G||$ si et seulement si $|F| \supseteq |G|$;

iv) Soient une formule G close et F une formule dont les variables libres sont parmi $x_1, \ldots, x_n, X_1, \ldots, X_p$; alors

$$||\forall x_1 \ldots \forall x_n \forall X_1 \ldots \forall X_p (G \to F)|| = ||G \to \forall x_1 \ldots \forall x_n \forall X_1 \ldots \forall X_p F||.$$

Démonstration. Découle trivialement des définitions. □

Définition 2.1.21.
Etant donné un ensemble saturé de processus $\perp\!\!\!\perp$, on dit qu'un terme t réalise une formule close F si $t \in |F|$; ce qui sera également dénoté par l'expression $t \Vdash F$.

Remarques : La valeur de vérité de $\bot$ est la plus grande au sens de l'inclusion, puisque $||\bot|| = ||\forall X\ X|| = \Pi$. Un terme dans $|\bot|$ réalisera donc toutes les formules, c'est-à-dire que la formule $\forall X(\bot \to X)$ est réalisée par $\lambda x\ x$.
Par ailleurs, il y a une plus petite valeur de vérité, à savoir l'ensemble vide, qui sera notée $\top$. On a alors $|\top| = \Lambda_c$.

Lemme 2.1.22.
Si $\perp\!\!\!\perp$ n'est pas vide, alors l'ensemble $|\bot|$ ne l'est pas non plus.

Ce point est crucial ; un terme réalisant $\bot$ peut être vu comme un terme possédant tout ce dont il a besoin pour s'exécuter correctement dans n'importe quel contexte.

Démonstration du lemme 2.1.22. Etant donné un processus $t \star \pi$ dans $\perp\!\!\!\perp$, le terme $(\mathsf{k}_\pi)t$ est dans $|\bot|$ puisque pour toute pile ρ, $(\mathsf{k}_\pi)t \star \rho \succ t \star \pi$, ce qui donne le résultat considérant le caractère saturé de $\perp\!\!\!\perp$. □

Le lemme suivant, qui sera fréquemment utilisé par la suite, indique que la notion de réalisabilité est naturellement compatible avec le modus ponens ainsi qu'avec les règles d'introduction et d'élimination des quantificateurs universels.

Lemme 2.1.23.
On a les assertions suivantes :

i) Soient ξ et η des termes réalisant respectivement les formules $F \to G$ et F. Alors $(\xi)\eta \Vdash G$.

ii) Soit X une variable de prédicat d'arité k. Alors $\xi \Vdash \forall X F(X)$ si et seulement si $\xi \Vdash F(\Phi)$ quel que soit $\Phi \in \mathscr{P}(\Pi)^{\mathcal{N}^k}$.

iii) $\xi \Vdash \forall x F(x)$ si et seulement si $\xi \Vdash F(n)$ quel que soit $n \in \mathcal{N}$.

Démonstration. *ii*) et *ii*) découlent trivialement des définitions de $|\forall X F|$ et $|\forall x F|$. Pour démontrer *i*), on considère une pile $\pi \in ||G||$. On doit montrer que le processus $(\xi)\eta \star \pi$ est dans $\perp\!\!\!\perp$. Mais celui-ci se réduit en $\xi \star \eta \cdot \pi$, avec $\eta \cdot \pi$ qui est dans $||F \to G||$, ce qui assure que ce dernier processus est dans $\perp\!\!\!\perp$ et donne le résultat par saturation de cet ensemble. □

On indique encore le résultat suivant, qui montre que la flèche est covariante à droite et contravariante à gauche du point de vue des valeurs de vérité.

Lemme 2.1.24.
Soient F, F', G et G' des formules closes.
i) *Si $||F|| \supseteq ||F'||$ et $||G|| \subseteq ||G'||$, alors $||F \to G|| \subseteq ||F' \to G'||$.*
ii) *Si $||F|| = ||F'||$ et $||G|| = ||G'||$, alors $||F \to G|| = ||F' \to G'||$.*

Démonstration. *i*) On considère une pile π dans $||F \to G||$. Celle-ci est donc de la forme $t \cdot \rho$ avec $\rho \in ||G|| \subseteq ||G'||$ et $t \in |F|$. Mais le lemme 2.1.20 assure qu'alors $t \in |F'|$, ce qui donne $t \cdot \rho \in ||F' \to G'||$.
Le point *ii*) se démontre de même. □

On peut maintenant énoncer le résultat suivant, qui exprime que la notion de réalisabilité est compatible avec la déduction en logique classique du second ordre.

Lemme 2.1.25 (lemme d'adéquation).
Soit un terme t dans L, des formules closes $F_1, \ldots, F_k$ et G telles que le séquent $x_1 : F_1, \ldots, x_k : F_k \vdash t : G$ soit dérivable à l'aide des règles de typage données en 2.1.16. Alors, quels que soient les termes $t_1, \ldots t_k$ tels que $t_i \Vdash F_i$ pour $1 \leqslant i \leqslant k$, on a $t[t_1/x_1, \ldots, t_k/x_k] \Vdash G$.

Si l'on prend pour G une formule close, on obtient que les démonstrations effectuées en logique classique du second ordre donnent des termes (ne contenant pas de continuation) qui réalisent le théorème considéré, et ce *indépendamment de l'ensemble $\perp\!\!\!\perp$ choisi* (d'où la terminologie de *quasi-preuve*). Il faut encore noter que ce résultat ne dépend pas des instructions dont on suppose disposer, sous réserve que l'on se soit donné l'instruction cc.

Le lemme d'adéquation découle trivialement du résultat suivant, adapté à une preuve par induction.

Lemme 2.1.26.
Soient $F_1, \dots F_k$ et G des formules usuelles, dont les variables libres sont parmi $y_1, \dots, y_m$, $Y_1, \dots Y_n$, tels que le prédicat Y_i soit d'arité k_i.
Supposons que le séquent $x_1 : F_1, \dots, x_k : F_k \vdash t : G$ soit dérivable avec les règles données en 2.1.16.
Soient une suite $(b_i)_{1 \leqslant i \leqslant m}$ d'éléments de $\mathscr{N}$ ainsi qu'une suite $(\Phi_i)_{1 \leqslant i \leqslant n}$ telle que $\Phi_i \in \mathscr{P}(\Pi)^{\mathscr{N}^{k_i}}$.
Si l'on considère une suite de termes $(t_i)_{1 \leqslant i \leqslant k}$ telle que
$t_i \Vdash F_i[b_1/y_1, \dots, b_m/y_m, \Phi_1/Y_1, \dots, \Phi_n/Y_n]$ pour tout $\perp\!\!\!\perp$, alors
$t[t_1/x_1, \dots, t_k/x_k] \Vdash G[b_1/y_1, \dots, b_m/y_m, \Phi_1/Y_1, \dots, \Phi_n/Y_n]$ pour tout $\perp\!\!\!\perp$.

Démonstration. On note Γ le contexte $x_1 : F_1, \dots, x_k : F_k$. La preuve se fait par induction sur la longueur de la dérivation de $\Gamma \vdash t : G$. Etant donné un terme u, on notera u' le terme $u[t_1/x_1, \dots, t_k/x_k]$; étant donnée une formule A, on notera A' la formule $A[b_1/y_1, \dots, b_m/y_m, \Phi_1/Y_1, \dots, \Phi_n/Y_n]$. On considère la dernière règle utilisée dans la dérivation.

S'il s'agit de la règle 1, c'est trivial.

S'il s'agit de la règle 2, t s'écrit $(u)v$, et les séquents $\Gamma \vdash u : A \to G$ et $\Gamma \vdash v : A$ sont dérivables. On doit montrer $(u')v' \Vdash G'$; c'est-à-dire que quelle que soit la pile π dans $||G'||$, le processus $(u')v' \star \pi$ est dans $\perp\!\!\!\perp$. Mais considérant que $\perp\!\!\!\perp$ est clos par antéréduction, il suffit de montrer qu'il contient le processus $u' \star v' \cdot \pi$. Cela découle du fait que d'une part $v' \cdot \pi \in ||A' \to G'||$, puisque par hypothèse d'induction $v' \in |A'|$, et d'autre part $u' \in |A' \to G'|$, toujours par hypothèse d'induction.

S'il s'agit de la règle 3, t s'écrit $\lambda x\ u$, G s'écrit $A \to B$, et le séquent $\Gamma, x : A \vdash u : B$ est dérivable. On doit alors montrer $\lambda x\ u' \Vdash A' \to B'$; c'est-à-dire que quelle que soit la pile π dans $||A' \to B'||$, le processus $\lambda x\ u' \star \pi$ est dans $\perp\!\!\!\perp$. Mais π est de la forme $v \cdot \rho$, avec v réalisant A' et ρ dans $||B'||$, et il suffit (comme $\perp\!\!\!\perp$ est saturé) de vérifier que le processus $u'[v/x] \star \rho$ est dans $\perp\!\!\!\perp$. Mais l'hypothèse d'induction assure $u'[v/x] \Vdash B'$, ce qui donne le résultat.

S'il s'agit de la règle 4, t s'écrit $(\mathrm{cc})u$ et le séquent $\Gamma \vdash u : (A \to B) \to A$ est dérivable. On doit alors montrer $(\mathrm{cc})u' \Vdash A'$; c'est-à-dire quelle que soit la pile $\pi \in ||A'||$, le processus $(\mathrm{cc})u' \star \pi$ est dans $\perp\!\!\!\perp$. On considère encore une fois un réduit de ce processus, à savoir $u' \star k_\pi \cdot \pi$. L'hypothèse d'induction assurant que u' réalise $(A' \to B') \to A'$, il suffit pour obtenir le résultat de démontrer que k_π réalise $A' \to B'$. Considérons donc un terme v réalisant A' ainsi qu'une pile ρ dans $||B'||$. Le processus $k_\pi \star v \cdot \rho$ est clairement dans $\perp\!\!\!\perp$, puisqu'il se réduit en $v \star \pi$ avec $\pi \in ||A'||$.

S'il s'agit de la règle 5, on G s'écrit $\forall x A$ et le séquent $\Gamma \vdash t : A$ est dérivable. L'hypothèse d'induction donne $t' \Vdash A'[n/x]$ pour tout élément

n de $\mathscr{N}$, puisque x n'est libre dans aucune des formules de Γ. Donc $t' \in \bigcap_{n\in\mathscr{N}} |A'[n/x]| = |\forall x A'| = |G'|$, ce qui donne le résultat.

S'il s'agit de la règle 6, on a G s'écrit $\forall X A$, avec X un prédicat d'arité k, et le séquent $\Gamma \vdash t : A$ est dérivable. L'hypothèse d'induction donne $t' \Vdash A'[R/X]$ pour tout élément R de $\mathscr{P}(\Pi)^{\mathscr{N}^k}$. On a donc $t' \in \bigcap_{R\in\mathscr{P}(\Pi)^{\mathscr{N}^k}} |A'[R/X]| = |G'|$.

S'il s'agit de la règle 7, G s'écrit $A[\tau/x]$ et le séquent $\Gamma \vdash t : \forall x A$ est dérivable. On a par hypothèse d'induction $t' \Vdash \forall x A'$. Mais si n est la valeur de τ, on a $|A'[\tau/x]| = |A'[n/x]| \supseteq |\forall x A'|$, ce qui donne le résultat.

Enfin s'il s'agit de la règle 8, G s'écrit $A[F(z_1,\dots,z_p)/X(z_1,\dots,z_p)]$ et le séquent $\Gamma \vdash t : \forall X A$ dérivable. On doit montrer que t' est dans $|A'[F'(z_1,\dots,z_p)/X(z_1,\dots,z_p)]|$, sachant que par hypothèse de récurrence on a $t' \in |\forall X A'|$; le résultat découle alors du lemme suivant en raisonnant comme dans le cas de la règle 7.

□

Lemme 2.1.27.
Soit F et A des formules à paramètres dont les seules variables libres sont $z_1,\dots,z_p$ (resp. X d'arité p). On définit un élément $R \in \mathscr{P}(\Pi)^{\mathscr{N}^p}$ par :

$$R(n_1,\dots,n_p) = ||F(n_1/z_1,\dots,n_p/z_p)|| \quad \textit{pour } n_1,\dots,n_p \textit{ dans } \mathscr{N}$$

Alors $||A[F/X(z_1,\dots,z_p)]|| = ||A[R/X]||$.

Démonstration. On prouve le résultat par induction sur la longueur de A. Les seules difficultés apparaissent lorsque A commence par un quantificateur universel.
Si A s'écrit $\forall x B$, on peut supposer que x n'est pas parmi $z_1,\dots z_p$ et alors :
$||A[F/X(z_1,\dots,z_p)]|| = ||\forall x B[F/X(z_1,\dots,z_p)]||$
$= \bigcup_{n\in\mathscr{N}} ||B[F/X(z_1,\dots,z_p)][n/x]||$
$= \bigcup_{n\in\mathscr{N}} ||B[n/x][F/X(z_1,\dots,z_p)]||$
$= \bigcup_{n\in\mathscr{N}} ||B[n/x][R/X]||$ par hypothèse d'induction
$= \bigcup_{n\in\mathscr{N}} ||B[R/X][n/x]|| = ||\forall x B[R/X]||$, ce qui donne le résultat dans ce cas.
Enfin si A s'écrit $\forall Y B$, avec Y un prédicat d'arité q que l'on peut supposer différent de X, on a :
$||A[F/X(z_1,\dots,z_p)]|| = \bigcup_{P\in\mathscr{P}(\Pi)^{\mathscr{N}^q}} ||B[F/X(z_1,\dots,z_p)][P/Y]||$
$= \bigcup_{P\in\mathscr{P}(\Pi)^{\mathscr{N}^q}} ||B[P/Y][F/X(z_1,\dots,z_p)]||$
$= \bigcup_{P\in\mathscr{P}(\Pi)^{\mathscr{N}^q}} ||B[P/Y][R/X]||$ par hypothèse d'induction
$= \bigcup_{P\in\mathscr{P}(\Pi)^{\mathscr{N}^q}} ||B[R/X][P/Y]|| = ||\forall Y B[R/X]||$, ce qui permet de conclure. □

2.1.5 Les modèles de la réalisabilité

Etant donné un ensemble saturé $\perp\!\!\!\perp$, l'ensemble $\mathscr{P}(\Pi)$ peut être muni d'un préordre qui en fait une algèbre de Boole dont la structure dépend de la notion de réalisabilité.

Définition 2.1.28.
Un ensemble saturé $\perp\!\!\!\perp$ sera dit cohérent s'il n'existe aucune quasi-preuve ξ tel que $\xi \Vdash \bot$ pour ce $\perp\!\!\!\perp$.

Théorème 2.1.29.
Pour U et V dans $\mathscr{P}(\Pi)$ et $\perp\!\!\!\perp$ saturé, notons $U \leq V$ la proposition suivante :

$$\exists \xi \in QP,\ \xi \Vdash U \to V$$

Si $\perp\!\!\!\perp$ est cohérent, la relation $\leq$ est un préordre sur $\mathscr{P}(\Pi)$ qui en fait une algèbre de Boole.

Démonstration. On considère un ensemble saturé cohérent $\perp\!\!\!\perp$. Tout au long de cette démonstration, les lettres U, V et W désignent des variables propositionnelles, c'est-à-dire des éléments de $\mathscr{P}(\Pi)$.
$\leq$ est un préordre sur $\mathscr{P}(\Pi)$: considérons avoir les relations $U \leq V$ et $V \leq W$, c'est-à-dire avoir deux quasi-preuves ξ et ξ' tels que $\xi \Vdash U \to V$ et
$\xi' \Vdash V \to W$. Il est alors clair que $\lambda x(\xi')(\xi)x$ (qui est encore une quasi-preuve) réalise $U \to W$; c'est-à-dire que $U \leq W$. En effet, une pile π dans $U \to W$ est de la forme $t \cdot \rho$, avec $t \in |U|$ et $\rho \in ||W||$, et il nous faut montrer que le processus $\lambda x(\xi')(\xi)x \star t \cdot \rho$ est dans $\perp\!\!\!\perp$. Or celui-ci se réduit en $\xi' \star (\xi)t \cdot \pi$, il nous suffit donc de démontrer que $(\xi)t$ réalise V, ce qui découle du lemme 2.1.23.
Axiome de la borne supérieur : On peut définir $Sup(U, V)$ par $U \vee V$. En effet, les trois formules $[U \to (U \vee V)]$, $[V \to (U \vee V)]$ ainsi que
$[(U \to Z), (V \to Z) \to ((U \vee V) \to Z)]$ sont démontrables, ce qui fournit (via le lemme 2.1.26) des quasi-preuve réalisant ces formules, et prouve donc que $U \vee V = Sup(U, V)$.
Axiomes de la borne inférieure : on peut de même définir $Inf(U, V)$ par $U \wedge V = \forall X[(U, V \to X) \to X]$.
Existence d'un plus petit élément : $\lambda x\ x \Vdash \bot \to U$, puisqu'un terme réalisant $\bot$ s'oppose à toutes les piles, et réalise donc n'importe quelle formule. $\bot$ est donc le plus petit élément de l'algèbre.
Existence d'un plus grand élément : $\lambda x\ x \Vdash U \to \top$, puisque $||U \to \top|| = \top$. Le plus grand élément de l'algèbre est donc $\top$.
Existence du complémentaire : Le complémentaire de U est donné par $\neg U$, puisque les formules $\neg U \wedge U \to \forall X X$ et $\neg U \vee U$ sont démontrables, ce qui assure que les énoncés $\neg U \wedge U \leq \bot$ et $\neg U \vee U \geq \top$ sont vrais.
Enfin, comme $\perp\!\!\!\perp$ est cohérent, il n'existe pas de quasi-preuve réalisant $\top \to \bot$ (car si ξ en était une, $(\xi)\lambda x\ x$ réaliserait $\bot$), ce qui assure que l'on n'a pas $\top \leq \bot$. □

Une fois fixé un ensemble $\perp\!\!\!\perp$ cohérent, chaque formule close prend une valeur booléenne. L'ensemble des formules qui prennent la valeur 1, c'est-à-dire celles qui sont réalisées par une quasi-preuve, est une théorie consistante qui contient celle des formules valides de la logique du second ordre. On appellera *modèle de la réalisabilité* ou *modèle générique* un modèle d'une telle théorie.

Notons déjà que le cas où $\perp\!\!\!\perp = \emptyset$ est trivial : on peut vérifier qu'alors l'algèbre de Boole est $\{0,1\}$, et que les modèles obtenus sont élémentairement équivalents au modèle de départ $\mathscr{N}$.

2.2 Quelques résultats fondamentaux

Cette partie détaille certains des résultats donnés dans [18], et indique les méthodes usuelles de raisonnement dans le cadre de la réalisabilité classique. On commence par donner la définition suivante, qui permet de construire les ensembles saturés qui seront utilisés pour obtenir des spécifications.

Définition 2.2.1.
Etant donné un processus p, on appelle « fil engendré par p » l'ensemble

$$\{q \in \Lambda_c \star \Pi; p \succ q\}.$$

On notera alors $< p >$ le complémentaire de cet ensemble.

Théorème 2.2.2.
Pour tout processus p, $< p >$ est saturé pour toutes les instructions.

Démonstration. Soit $q \in< p >$, c'est-à-dire tel que p ne se réduise pas en q. Soit encore q' tel que $q' \succ q$. Si $q' \notin< p >$ alors $p \succ q'$, et donc $p \succ q$ par transitivité de $\succ$, ce qui est une contradiction. □

2.2.1 Un premier exemple de spécification

Théorème 2.2.3.
Soit θ un terme réalisant la formule $\forall X(X \to X)$ quel que soit l'ensemble $\perp\!\!\!\perp$ saturé. Alors quels que soient $t \in \Lambda_c$ et $\pi \in \Pi$, $\theta \star t \cdot \pi \succ t \star \pi$.

Remarques :
- Le lemme d'adéquation assure que la conclusion reste valable pour un terme ξ possédant le type $\forall X(X \to X)$, ce qui est le cas des termes issus d'une preuve en logique classique de cette formule. Ce résultat indique que tous les termes réalisant la formule $\forall(X \to X)$ se comportent comme l'identité $\lambda x\ x$.
- Les spécifications se démontrent toujours en choisissant un $\perp\!\!\!\perp$ particulier. Si les ensembles saturés cohérents servent à donner des modèles à la réalisabilité, il est à noter que ce sont systématiquement les ensembles *incohérents* qui donnent des spécifications.
- Il est encore à noter que la preuve qui suit ne suppose rien sur la nature des instructions disponibles, c'est-à-dire sur les axiomes dont on s'est servi pour démontrer la formule considérée. Ce fait est extrêmement important, car il nous permet d'ajouter de nouvelles instructions sans modifier le problème de la spécification.

On va donner deux preuves de ce résultat, qui diffèrent par le choix de l'ensemble $\perp\!\!\!\perp$: dans la première, on choisit pour $\perp\!\!\!\perp$ le processus considéré et ses réduits, alors que dans la seconde, on considère l'ensemble des processus qui

possède la terminaison voulue. On pourra presque toujours choisir arbitrairement l'une de ces deux méthodes.

Première démonstration. t et π étant donnés, on prend $\perp\!\!\!\perp =< \theta \star t \cdot \pi >$. On appelle Φ le singleton $\{\pi\}$; on a $\theta \Vdash \Phi \to \Phi$. Mais alors comme $\theta \star t \cdot \pi \notin \perp\!\!\!\perp$, il vient $t \cdot \pi \notin ||\Phi \to \Phi||$, ce qui donne $t \nVdash \Phi$ (puisque $\pi \in \Phi$) pour ce $\perp\!\!\!\perp$. On en déduit donc $t \star \pi \notin \perp\!\!\!\perp$, c'est-à-dire $\theta \star t \cdot \pi \succ t \star \pi$, ce qui prouve le résultat. □

Seconde démonstration. t et π étant donnés, soit $\perp\!\!\!\perp = \{p \in \Lambda_c \star \Pi \ ; \ p \succ t \star \pi\}$, qui est clairement saturé. On appelle encore Φ le singleton $\{\pi\}$. Le lemme 2.1.23 donne alors $\theta \Vdash \Phi \to \Phi$. Mais comme $t \Vdash \Phi$ (puisque $t \star \pi \in \perp\!\!\!\perp$), on a $t \cdot \pi \in ||\Phi \to \Phi||$. On en déduit $\theta \star t \cdot \pi \in \perp\!\!\!\perp$, ce qui donne le résultat.
□

Si $Bool(x)$ désigne la formule $\forall X(X(1), X(0) \to X(x))$, on démontrerait de même le résultat suivant.

Théorème 2.2.4.
Soit ξ un terme réalisant la formule $Bool(1)$ quel que soit l'ensemble $\perp\!\!\!\perp$ saturé. Alors quels que soient $t, u \in \Lambda_c$ et $\pi \in \Pi$, $\xi \star t \cdot u \cdot \pi \succ t \star \pi$.

2.2.2 L'égalité

En règle général, la valeur de vérité d'une formule dépend de l'ensemble $\perp\!\!\!\perp$ choisi et n'est pas clairement explicitable. On peut néanmoins faire ce travail pour les formule de la forme $\tau = \tau'$.

Théorème 2.2.5.
Soient τ et τ' des éléments de $T_{\mathscr{N}}$. Alors

$$||\tau = \tau'|| = \begin{cases} ||\forall X(X \to X)|| = \{t \cdot \pi; t \star \pi \in \perp\!\!\!\perp\} \text{ si } \tau \text{ et } \tau' \text{ ont la même valeur,} \\ ||\top \to \bot|| = \{t \cdot \pi; t \in \Lambda_c, \pi \in \Pi\} \text{ sinon.} \end{cases}$$

Démonstration. Par définition, on a

$$||\tau = \tau'|| = ||\forall X(X\tau \to X\tau')|| = \bigcup_{\Phi \in \mathscr{P}(\Pi)^{\mathscr{N}}} ||\Phi\tau \to \Phi\tau'||.$$

Si τ et τ' ont la même valeur n, alors considérant que $||\Phi\tau|| = ||\Phi\tau'|| = \Phi n$ parcourt $\mathscr{P}(\Pi)$ lorsque Φ parcourt $\mathscr{P}(\Pi)^{\mathscr{N}}$, on en déduit

$$||\tau = \tau'|| = \bigcup_{\Psi \in \mathscr{P}(\Pi)} ||\Psi \to \Psi|| = ||\forall X(X \to X)||.$$

Enfin, si τ et τ' prennent des valeurs différentes, on obtient en prenant Φ tel que $||\Phi\tau|| = \top$ et $||\Phi\tau'|| = \Pi$ que $||\tau = \tau'|| \supseteq ||\top \to \bot||$. Considérant que $\top$ (resp. $\bot$) est la plus petite (resp. plus grande) valeur de vérité, on conclut en utilisant le lemme 2.1.24. □

Les résultats suivant découlent trivialement de ce calcul. Il faut noter que les symboles de fonction considérés sont quelconques : on ne leur demande pas d'être associé à des fonctions récursives.

Théorème 2.2.6.
Si la fonction k-aire f est telle que $\mathscr{N} \vDash \forall x_1 \dots \forall x_k \, f(x_1, \dots, x_k) = 0$, alors la formule $\forall x_1 \dots \forall x_k (f(x_1, \dots, x_k) = 0)$ est réalisée par I.

La spécification de ces formules est donnée par le théorème 2.2.3 : tout terme réalisant un théorème de cette forme se comporte comme $\lambda x \; x$.

Théorème 2.2.7.
Soit F une formule de la forme $\forall x \exists y \; f(x, y) = 0$ ou $\exists x \forall y \; f(x, y) = 0$, f étant un symbole de fonction 2-aire. Si $\mathscr{N} \vDash F$, alors $\lambda x(x) I \Vdash F$.

Dans un modèle de la réalisabilité, chaque fonction $f : \mathscr{N}^k \to \mathscr{N}$ possède une interprétation. Ce théorème indique par exemple que l'interprétation d'un symbole de fonction surjective est une application surjective.

Théorème 2.2.8.
Si les fonctions k-aires f et g sont telles que

$$\mathscr{N} \vDash \forall x_1 \dots \forall x_k (f(x_1, \dots, x_k) = 0 \to g(x_1, \dots, x_k) = 0),$$

alors cette formule est réalisée par la quasi-preuve I.

En outre, on en déduit que l'interprétation dans un modèle de la réalisabilité d'un symbole de fonction injective est une application injective. Néanmoins, on démontrera dans la suite (Cf le théorème 2.2.31) que, en général, l'application interprétant le symbole f ne laisse pas stable l'ensemble des entiers du modèle.

Ces résultats se généralisent de la manière suivante.

Théorème 2.2.9.
Supposons que

$$\mathscr{N} \vDash \forall x_1 \exists y_1 \dots \forall x_p \exists y_p (A_0, \dots, A_n \to A_{n+1}),$$

où chacun des A_i est de la forme $f_i(x_1, y_1, \dots, x_p, y_p) = 0$. Alors cette formule est réalisée par la quasi-preuve

$$\lambda x_1(x_1) \dots \lambda x_p(x_p) \lambda y_0 \dots \lambda y_n(y_0) \dots (y_n) I.$$

Les modèles de la réalisabilité sont donc élémentairement équivalents au modèle de départ en ce qui concerne les formules du premier ordre écrites avec l'égalité comme seule formule atomique.

2.2.3 Un prédicat pour la négation de l'égalité

Définition 2.2.10.
On appelle $\neq$ la constante de prédicat binaire définie par

$$n \neq n' = \begin{cases} \Pi \text{ si } n = n' \\ \top \text{ sinon.} \end{cases}$$

Le théorème suivant montre que l'on peut avantageusement utiliser la formule $x \neq y$ à la place de $x = y \to \bot$; on simplifiera ainsi la valeur de vérité des formules à réaliser, ce qui permettra d'en simplifier les réalisateurs.

Théorème 2.2.11.
On a les assertions suivantes :

1. $\lambda x(x)I \Vdash \forall x \forall y[(x = y \to \bot) \to x \neq y]$
2. $\lambda x \lambda y(y)x \Vdash \forall x \forall y[x \neq y \to (x = y \to \bot)]$

Démonstration. 1. La définition de $\neq$ assure d'une part que si n et m sont différents, alors

$$||(n = m \to \bot) \to n \neq m|| = ||(n = m \to \bot) \to \top|| = \top$$

et d'autre part que pour tout $n \in \mathcal{N}$

$$||(n = n \to \bot) \to n \neq n|| = ||(n = n \to \bot) \to \bot||.$$

Il suffit donc de vérifier $\lambda x\ (x)I \Vdash (\forall X(X \to X) \to \bot) \to \bot$.
Soit une pile π quelconque, et t un terme réalisant $\forall X(X \to X) \to \bot$. On veut montrer que $\lambda x\ (x)I \star t \cdot \pi$ est dans $\perp\!\!\!\perp$. Or ce processus se réduit en $t \star I \cdot \pi$, qui est dans $\perp\!\!\!\perp$ puisque $I \Vdash \forall X(X \to X)$. On en déduit le résultat par saturation de $\perp\!\!\!\perp$.

2. On peut encore une fois calculer la valeur de vérité de cette formule, qui est égale à$||\top, (\top \to \bot) \to \bot|| \bigcup ||\bot, \forall X(X \to X) \to \bot||$.
Soient alors u et v des termes réalisant respectivement $\top$ et $\top \to \bot$, π une pile quelconque. On doit montrer $\lambda x \lambda y(y)x \star u \cdot v \cdot \pi \in \perp\!\!\!\perp$. Mais ce processus se réduit en $v \star u \cdot \pi$, avec $u \cdot \pi \in ||\top \to \bot||$; ce dernier processus est donc dans $\perp\!\!\!\perp$, et celui-ci étant saturé on en déduit que $\lambda x \lambda y(y)x$ est dans $|\top, (\top \to \bot) \to \bot|$.
Soient enfin u et v réalisant respectivement $\bot$ et $\forall X(X \to X)$, π une pile quelconque. On doit montrer que le processus $\lambda x \lambda y(y)x \star u \cdot v \cdot \pi$ est dans $\perp\!\!\!\perp$. Celui-ci se réduit en $v \star u \cdot \pi$. Il faut alors remarquer que $u \star \pi \in \perp\!\!\!\perp$, puisque $u \Vdash \bot$. On en déduit $u \cdot \pi \in ||\forall X(X \to X)||$, ce qui prouve $\lambda x \lambda y(y)x \in |\bot, \forall X(X \to X) \to \bot|$ et termine la démonstration.
□

De la définition de $\neq$ découlent trivialement les résultats suivants, analogues à ceux de la sous-section précédente.

Théorème 2.2.12.
Si la fonction k-aire f est telle que $\mathscr{N} \vDash \forall x_1 \ldots \forall x_k \; f(x_1, \ldots, x_k) \neq 0$, alors cette formule est réalisée par n'importe quelle quasi-preuve.

Théorème 2.2.13.
Soit F une formule de la forme $\forall x \exists y \; f(x,y) \neq 0$ ou $\exists x \forall y \; f(x,y) \neq 0$, f étant un symbole de fonction 2-aire. Si $\mathscr{N} \vDash F$, alors $\lambda x(x) I \Vdash F$.

Théorème 2.2.14.
Si la fonction k-aire f est telle que

$$\mathscr{N} \vDash \forall x_1 \ldots \forall x_k (f(x_1, \ldots, x_k) \neq 0 \to g(x_1, \ldots, x_k) \neq 0),$$

alors cette formule est réalisée par I.

Néanmoins, on verra dans la suite que la formule

$$\forall x (x \neq 0, x \neq 1 \to x^2 \neq x)$$

n'est pas réalisable indépendamment de l'ensemble $\perp\!\!\!\perp$ choisi. Les modèles de la réalisabilité ne sont donc pas élémentairement équivalents au modèle de départ.

2.2.4 Conservation de la bonne fondation

On fixe pour la fin de cette sous-partie un ensemble $\mathcal{N}$ qui est bien fondé pour une application φ injective ; c'est-à-dire que chaque élément peut s'écrire de manière unique $\varphi^n z$, avec n entier et z qui vérifie $\forall y(\varphi y \neq z)$.
On peut par exemple prendre pour $\mathcal{N}$ le modèle standard de l'arithmétique $\mathbb{N}$, et pour φ le successeur. Dans le cas général, on a affaire à des ensembles dans lesquels il existe plusieurs éléments ne possédant pas d'antécédent pour l'application φ. Un tel ensemble est représenté à la figure 2.1.

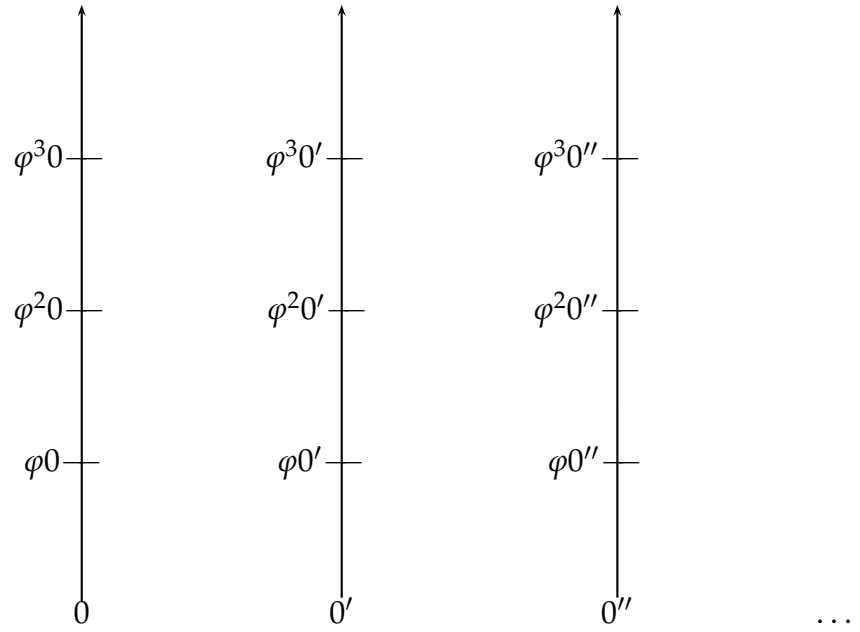


FIG. 2.1 – Un ensemble bien fondé pour une relation $y = \varphi(x)$ avec φ injective.

Chacun des fils verticaux est de hauteur ω ; le nombre d'élément ne possédant pas d'antécédent pour φ est quelconque (il n'est pas nécessairement dénombrable).

Définition 2.2.15.
On note $\lhd$ la relation bien fondée sur $\mathcal{N}$ définie par :

$$x \lhd y \text{ ssi } \exists m \in \mathbb{N}^*, \varphi^m(x) = y.$$

Définition 2.2.16.
On note Y le combinateur de point fixe de Turing défini par $Y = (A)A$, où A désigne la quasi-preuve $\lambda a \lambda b(b)(a)ab$.

On rappelle que la réduction de tête de Y donne lieu à l'exécution suivante, quels que soient $t \in \Lambda_c$ et $\pi \in \Pi$:

$$Y \star t \cdot \pi \succ t \star (Y)t \cdot \pi$$

On dispose du résultat suivant, qui exprime que Y assure la conservation de la bonne fondation sur les modèles génériques ; c'est-à-dire que la structure représentée à la figure 2.1 est conservée par la réalisabilité.

Théorème 2.2.17.
Soit $\psi : \mathcal{N}^2 \to \mathcal{N}$ telle que $\psi(y,x) = 0$ définisse une relation bien fondée sur $\mathcal{N}^2$. Alors Y réalise la formule

$$\forall X\{\forall x[\forall y(Xy \to \psi(y,x) \neq 0) \to \neg Xx] \to \forall z \neg Xz\}.$$

En prenant ψ telle que $\psi(y,x) = x - \varphi(y)$, on obtient que Y réalise

$$\forall X\{\forall x[\forall y(Xy \to \varphi y \neq x) \to \neg Xx] \to \forall z \neg Xz\}.$$

Le lemme d'adéquation assure alors que la formule suivante, qui lui est équivalente en logique classique du second ordre, est réalisée :

$$\forall X\{\forall x[\forall y(\varphi y = x \to Xy) \to Xx] \to \forall z Xz\}.$$

Les modèles génériques conservent donc les propriétés de bonne fondation des modèles de départ : on peut y effectuer des raisonnements par induction. Néanmoins, on verra dans la partie suivante que le nombre d'éléments qui ne sont pas des successeurs pour φ n'est pas conservé.

Démonstration du théorème 2.2.17. On fixe un ensemble saturé $\perp\!\!\!\perp$, ainsi que $R : \mathcal{N} \to \mathcal{P}(\Pi)$. On doit montrer que quel que soit le terme t réalisant $\forall x[\forall y(R(y) \to \psi(y,x) \neq 0) \to \neg R(x)]$, $z \in \mathcal{N}$ et la pile π dans $||R(z) \to \perp||$, le processus $Y \star t \cdot \pi$ est dans $\perp\!\!\!\perp$. On va démontrer ce résultat par induction sur z relativement à Ψ. Le processus considéré se réduit en $t \star (Y)t \cdot \pi$, avec t qui réalise $\forall y(R(y) \to \psi(y,z) \neq 0) \to \neg R(z)$.
Si z ne possède pas d'antécédent pour Ψ, on a $||\psi(y,z) \neq 0|| = \top$ pour tout $y \in \mathcal{N}$, et donc $t \Vdash \top \to \neg R(z)$, ce qui donne le résultat.
S'il existe z' tel que $\psi(z',z) = 0$, on doit montrer

$$(Y)t \in |\forall y(R(y) \to \psi(y,z) \neq 0)| = \bigcap_{\{z';\psi(z',z)=0\}} |R(z') \to \perp|,$$

ce qui est vrai par hypothèse d'induction. □

On n'utilisera pas directement ce résultat par la suite, car il mènerait à des réalisateurs trop complexes. On lui préférera les deux résultats suivants, dont les utilisations sont plus naturelles.

Théorème 2.2.18.

$$Y \Vdash \forall X \left(\forall x \left(\bigcap_{y \lhd x} \mid Xy \mid\!\!\rightarrow Xx \right) \rightarrow \forall z Xz \right).$$

Démonstration. On fixe $\Psi : \mathscr{N} \rightarrow \mathscr{P}(\Pi)$, $z \in \mathscr{N}$ et t un terme réalisant $\forall x(\bigcap_{y \lhd x} \mid \Psi y \mid\!\!\rightarrow \Psi x)$. On va montrer par induction sur z que $Y \star t \cdot \pi \in \perp\!\!\!\perp$ pour toute pile $\pi \in \Psi(z)$. Ce processus se réduit en $t \star (Y)t \cdot \pi$, avec t qui réalise $\bigcap_{y \lhd z} \mid \Psi y \mid\!\!\rightarrow \Psi z$.
Si z ne possède pas d'antécédent pour φ, on a $\bigcap_{y \lhd z} \mid \Psi y \mid = \Lambda_c$, et donc $t \Vdash \top \rightarrow \Psi(z)$, ce qui donne le résultat.
Si $z = \varphi(z')$, on doit montrer $(Y)t \in\mid \Psi z' \mid$, ce qui découle de l'hypothèse d'induction et termine la démonstration. □

Corollaire 2.2.19.
Supposons que l'ensemble de base $\mathscr{N}$ est le modèle standard de l'arithmétique. On appelle s le successeur sur les entiers.
Soit $F(x)$ une formule avec une seule variable libre x, et t un terme tel que :
- $t \Vdash \forall x(F(x) \rightarrow F(sx))$,
- $t \Vdash \top \rightarrow F(0)$.

Alors $(Y)t$ réalise $\forall x F(x)$.

Démonstration. Découle trivialement du théorème précédent. □

Remarque : La seule propriété de Y utilisée dans ces preuves est sa règle de réduction ; on peut donc également introduire une instruction notée Y qui possède cette règle de réduction, et l'utiliser comme réalisateur de la formule exprimant la bonne fondation. Cela possède l'avantage de simplifier les exécutions des processus où apparaît Y, mais on doit alors se restreindre aux ensembles $\perp\!\!\!\perp$ saturés pour cette instruction.

2.2.5 Ce qui est « vrai » n'est pas toujours réalisable

On suppose dans la suite avoir la propriété suivante quels que soient les processus p, q et q' :

$$p \succ q, p \succ q', q \not\succ q' \implies q' \succ q$$

Il s'agit d'une restriction faite sur les instructions disponibles : on suppose que chacune d'entre elle possède au plus un réduit au cours d'une exécution donnée.

Théorème 2.2.20.
Il n'existe aucun terme dans $|\top, \bot \rightarrow \bot| \cap |\bot, \top \rightarrow \bot|$ quel que soit $\perp\!\!\!\perp$.

Démonstration. Supposons avoir un terme $\xi \in |\top, \bot \to \bot| \cap |\bot, \top \to \bot|$ pour tout $\perp\!\!\!\perp$. On pose $\delta = \lambda x(x)x$, $\Omega_0 = (\delta)\delta\underline{0}$ et $\Omega_1 = (\delta)\delta\underline{1}$, avec $\underline{0} = \lambda f\lambda x\ x$, et $\underline{1} = \lambda f\lambda x\ (f)x$. On fixe encore une pile π, et l'on considère le processus $q = \xi \star \Omega_1 \cdot \Omega_0 \cdot \pi$.
On prend d'abord $\perp\!\!\!\perp = \{p \in \Lambda_c \star \Pi\ ;\ p \succ \Omega_0 \star \rho,\ \rho \in \Pi\}$. Pour cet ensemble saturé, on a clairement $\Omega_0 \Vdash \bot$ et comme $\xi \Vdash \bot, \top \to \bot$, on en déduit $q \in \perp\!\!\!\perp$; c'est-à-dire $q \succ \Omega_0 \star \rho$ pour $\rho \in \Pi$.
On prend ensuite $\perp\!\!\!\perp = \{p \in \Lambda_c \star \Pi\ ;\ p \succ \Omega_1 \star \rho,\ \rho \in \Pi\}$. Dans ce cas on a clairement $\Omega_1 \Vdash \bot$ et comme $\xi \Vdash \top, \bot \to \bot$, on en déduit $q \in \perp\!\!\!\perp$; c'est-à-dire $q \succ \Omega_1 \star \rho'$ pour $\rho' \in \Pi$.
Chaque processus possédant au plus un réduit, on a par exemple affaire à une situation du type $q \succ \Omega_0 \star \rho \succ \Omega_1 \star \rho'$. Mais les seuls réduits du processus $\Omega_0 \star \rho$ sont $\Omega_0 \star \rho$, $(\delta)\delta \star \underline{0} \cdot \rho$ et $\delta \star \delta \cdot \underline{0} \cdot \rho$, ce qui donne une contradiction. □

On voit ici apparaître une différence fondamentale entre la réalisabilité et le forcing : l'intersection des valeurs de vérité de deux formules prouvables peut ne pas être réalisée par une quasi-preuve, ce qui est dû à l'absence dans Λ_c de terme effectuant le calcul d'une « disjonction parallèle ».
Ce résultat permet de donner un premier exemple de formule qui n'est pas, en général, réalisable.

Corollaire 2.2.21.
La formule $\forall x(x \neq 0, x \neq 1 \to x^2 \neq x)$ *n'est pas réalisée indépendamment de* $\perp\!\!\!\perp$.

Il y a donc, en général, des booléens non-standards dans les modèles génériques. On se reportera à [18] où est donné un ensemble saturé $\perp\!\!\!\perp$ pour lequel la négation de cette formule est réalisée.

Démonstration. On vérifie que

$$|\forall x(x \neq 0, x \neq 1 \to x^2 \neq x)| = |\top, \bot \to \bot| \cap |\bot, \top \to \bot|.$$

Il suffit alors d'appliquer le théorème précédent. □

2.2.6 Les entiers comme type de données

Notations : On notera s le successeur sur les entiers. On notera par le même symbole un terme dans $T_{\mathcal{M}}$ et sa valeur.
On note encore $\underline{n}$ l'entier de Church associé à l'entier n, c'est-à-dire le terme $\lambda x\lambda f(f)^n x$.
On dénote enfin $Ent(x)$ la formule $\forall X\{\forall y(Xy \to Xs(y)), X0 \to Xx\}$.

On considère ici comme ensemble de base le modèle standard de l'arithmétique, $\mathbb{N}$. Nous allons étudier les propriétés du type « entier », et montrer que

la réalisabilité permet d'assurer la correction des calculs sur celui-ci.

Lemme 2.2.22.
Pour tout entier n, $\underline{n} \Vdash Ent(s^n 0)$.

Démonstration. Soit $n \in \mathbb{N}$. On dénote par Γ le contexte

$$f : \forall i(Xi \to Xsi), x : X0$$

Il suffit de considérer la dérivation suivante et d'appliquer le lemme d'adéquation.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash x : X0 \qquad \Gamma \vdash f : X0 \to Xs0}{\Gamma \vdash (f)x : Xs0} \qquad \Gamma \vdash f : Xs0 \to Xs^2 0}{\Gamma \vdash (f)^2 x : Xs^2 0}}{\vdots}}{\Gamma \vdash (f)^n x : Xs^n 0}}{f : \forall i(Xi \to Xsi) \vdash \lambda x (f)^n x : X0 \to Xs^n 0}}{\vdash \lambda f \lambda x (f)^n x : \forall i(Xi \to Xsi), X0 \to Xs^n 0}}{\vdash \lambda f \lambda x (f)^n x : \forall X[\forall i(Xi \to Xsi), X0 \to Xs^n 0]}$$

□

Citons encore le résultat suivant, prouvé dans [18].

Théorème 2.2.23.
Soit ξ un λ-terme clos tel que $\xi \simeq_\beta \underline{n}$; alors $\xi \Vdash Ent(s^n 0)$.

L'arithmétique du second ordre

Les axiomes de l'arithmétique de Peano du second ordre peuvent être regroupés en quatre classes :

1. Des formules équationnelles : $p0 = 0$; $\forall x(psx = x)$; $\forall x(x + 0 = x)$; $\forall x \forall y(x + sy = s(x + y))$; $\forall x(x.0 = 0)$; $\forall x \forall y(x.sy = x.y + x)$.
2. Une formule inéquationnelle $\forall x(0 \neq sx)$.
3. L'axiome de récurrence $\forall x\ Ent(x)$.
4. L'axiome du choix dépendant.

Le théorème 2.2.6 assure que les axiomes de 1 sont réalisés par I ; le théorème 2.2.14 que l'axiome 2 est réalisé par n'importe quel quasi-preuve. Le cas de l'axiome du choix dépendant est traité dans [18] ; nous verrons dans le chapitre 6 comment réaliser une formule exprimant un axiome du choix dépendant restreint aux individus. Mais contrairement à la bonne fondation qui

est conservée dans les modèles génériques, nous allons voir que l'axiome de récurrence $\forall x Ent(x)$ n'est pas réalisé.
On reprend l'hypothèse faite sur l'exécution des processus à la page 28.

Théorème 2.2.24.
Il n'existe aucun terme qui réalise $\forall x\ Ent(x)$ quel que soit l'ensemble saturé $\perp\!\!\!\perp$.

Il y a donc (en général) dans les modèles génériques des éléments qui ne sont pas des entiers. On se reportera par ailleurs à [18], où est exhibé un modèle générique dans lequel il existe des éléments non-standards ainsi que des entiers non-standards.

Démonstration. Supposons avoir un terme ξ qui réalise $\forall x\ Ent(x)$ quel que soit l'ensemble saturé $\perp\!\!\!\perp$. On pose alors $\delta = \lambda x(x)x$, $\Omega_0 = (\delta)\delta\underline{0}$ et $\Omega_1 = (\delta)\delta\underline{1}$. On fixe encore une pile π, et l'on considère le processus $q = \xi \star \lambda x\ \Omega_1 \cdot \Omega_0 \cdot \pi$.
On prend d'abord $\perp\!\!\!\perp = \{p \in \Lambda_c \star \Pi\ ;\ p \succ \Omega_0 \star \pi\}$. On définit un prédicat unaire Φ en posant $\Phi(0) = \{\pi\}$ et $\Phi s(i) = \top$ pour tout entier i. Pour le $\perp\!\!\!\perp$ considéré, il est clair que $\lambda x\ \Omega_1 \Vdash \forall y(\Phi y \to \Phi s(y))$ et $\Omega_0 \Vdash \Phi 0$. Comme $\xi \Vdash Ent(0)$, il réalise en particulier $\forall y(\Phi y \to \Phi s(y)), \Phi 0 \to \Phi 0$ et l'on en déduit $q \in \perp\!\!\!\perp$; c'est-à-dire $q \succ \Omega_0 \star \pi$.
On prend ensuite $\perp\!\!\!\perp = \{p \in \Lambda_c \star \Pi\ ;\ p \succ \Omega_1 \star \pi\}$. On définit un prédicat unaire Ψ en posant $\Psi(0) = \top$ et $\Psi s(i) = \{\pi\}$ pour tout entier i. Pour le $\perp\!\!\!\perp$ considéré, il est clair que $\lambda x\ \Omega_1 \Vdash \forall y(\Psi y \to \Psi s(y))$ et $\Omega_0 \Vdash \Psi 0$. Comme $\xi \Vdash Ent(s0)$, il réalise en particulier $\forall y(\Psi y \to \Psi s(y)), \Psi 0 \to \Psi s0$ et l'on en déduit $q \in \perp\!\!\!\perp$; c'est-à-dire $q \succ \Omega_1 \star \pi$.
On peut alors conclure comme dans la preuve du théorème 2.2.20 : on a par exemple affaire à une situation du type $q \succ \Omega_0 \star \pi \succ \Omega_1 \star \pi$. Mais les seuls réduits du processus $\Omega_0 \star \pi$ sont $\Omega_0 \star \pi$, $(\delta)\delta \star \underline{0} \cdot \pi$ et $\delta \star \delta \cdot \underline{0} \cdot \pi$, ce qui donne une contradiction. □

On peut pallier la défection de l'axiome de récurrence en utilisant le résultat suivant :

Toute preuve d'une formule F en logique classique du second ordre utilisant les axiomes de 1, 2 et 3 cités précédemment peut être transformée en une preuve de F^{Ent} n'utilisant que les axiomes de 1 et 2 ainsi que le schéma suivant :
« $\forall x_1 \ldots \forall x_k[Ent(x_1), \ldots, Ent(x_k) \to Ent(f(x_1, \ldots, x_k))]$ pour chaque symbole de fonction f du langage considéré, »
F^{Ent} désignant la formule obtenue en restreignant tous les quantificateurs de F aux entiers.

La sous-section suivante donne un outil indispensable pour développer cette méthode. On verra ensuite que les fonctions pour lesquelles la formule ci-dessus est réalisée sont les fonctions récursives totales.

Opérateur de mise en mémoire

On trouvera une généralisation au cas d'un type de donnée quelconque des résultats de cette section dans [20].

Définition 2.2.25.
On note s la quasi-preuve $\lambda n \lambda f \lambda x (f)(n) f x$.

Théorème 2.2.26.
s réalise la formule $\forall x [Ent(x) \to Ent(sx)]$.

Démonstration. Le résultat découle encore de la conjugaison du lemme d'adéquation et de la dérivation qui suit ; où Γ désigne le contexte suivant :

$$n : Ent(s^n 0), f : \forall i (Xi \to Xsi), x : X0$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash n : \forall i (Xi \to Xsi), X0 \to Xs^n 0} \quad \dfrac{}{\Gamma \vdash f : \forall i (Xi \to Xsi)}}{\Gamma \vdash (n) f : X0 \to Xs^n 0}}{\Gamma \vdash (n) f : X0 \to Xs^n 0} \quad \dfrac{}{\Gamma \vdash x : X0}}{\Gamma \vdash (n) f x : Xs^n 0}}{\vdots}}{\dfrac{\dfrac{}{\Gamma \vdash f : Xs^n 0 \to Xs^{n+1} 0} \quad \dfrac{\dfrac{}{\vdots}}{\Gamma \vdash (n) f x : Xs^n 0}}{\Gamma \vdash (f)(n) f x : Xs^{n+1} 0}}}{n : Ent(s^n 0), f : \forall i (Xi \to Xsi) \vdash \lambda x (f)(n) f x : X0 \to Xs^{n+1} 0}}{n : Ent(s^n 0) \vdash \lambda f \lambda x (f)(n) f x : \forall i (Xi \to Xsi), X0 \to Xs^{n+1} 0}}{n : Ent(s^n 0) \vdash \lambda f \lambda x (f)(n) f x : Ent(s^{n+1} 0)}}{\vdash \lambda n \lambda f \lambda x (f)(n) f x : Ent(s^n 0) \to Ent(s^{n+1} 0)}}{\vdash \lambda n \lambda f \lambda x (f)(n) f x : \forall x [Ent(x) \to Ent(sx)]}$$

□

Définition 2.2.27.
On appelle T la quasi-preuve $\lambda f \lambda n ((n) \lambda g \lambda x (g)(s) x) f \underline{0}$. *Elle est appelée opérateur de mise en mémoire.*

On vérifie aisément que, quel que soit le λ-terme u, si ν désigne un λ-terme β-équivalent à un entier de Church $\underline{n}$, la réduction de tête faible du terme $(T)(u)\nu$ mène à $(u)s^n\underline{0}$. Le terme T permet donc d'évaluer l'argument ν avant d'évaluer le résultat de l'application de u à celui-ci, c'est-à-dire de simuler une stratégie d'appel par valeur dans un contexte d'exécution en appel par nom, d'où la dénomination d'*opérateur de mise en mémoire*.
Le théorème suivant donne une manière d'utiliser les opérateurs de mise en mémoire dans le cadre de la réalisabilité classique.

Théorème 2.2.28.
Soit n un entier. Si u est un terme et Φ un ensemble de pile tel que $u \star s^n\underline{0} \cdot \pi \in \perp\!\!\!\perp$ pour toute pile $\pi \in \Phi$, alors $(T)u \Vdash Ent(s^n0) \to \Phi$.

Ce résultat assure que lorsque l'on cherche à réaliser un théorème, on peut supposer quitte à rajouter *in fine* des opérateurs de mise en mémoire, que les seuls arguments du type $Ent(s^n0)$ que ce terme aura à traiter sont de la forme $s^n\underline{0}$.

Démonstration. Soit $\Phi \in \mathscr{P}(\Pi)$ et $\nu \Vdash Ent(s^n0)$. On doit montrer que le processus $(T)u \star \nu \cdot \pi$ est dans $\perp\!\!\!\perp$. On définit un prédicat unaire Ψ en posant :

$$\Psi(i) = \begin{cases} \{s^{n-i}\underline{0} \cdot \pi\} \text{ si } 0 \leqslant i \leqslant n \\ \top \text{ sinon} \end{cases}$$

L'hypothèse faite sur u assure $u \Vdash \Psi(0)$. Le processus considéré se réduisant en $\nu \star \lambda g\lambda x(g)(s)x \cdot u \cdot \underline{0} \cdot \pi$, il nous reste à démontrer
$\lambda g\lambda x(g)(s)x \Vdash \forall i[\Psi(i) \to \Psi(si)]$, ce qui donnera le résultat par l'hypothèse faite sur ν.
Soit un entier i, un terme $t \in |\Psi(i)|$ et une pile $\rho \in ||\Psi(si)||$; on doit montrer $\lambda g\lambda x(g)(s)x \star t \cdot \rho \in \perp\!\!\!\perp$. On peut supposer que $i < n$, puisque une telle pile ρ n'existe pas sinon. Dans ce cas $\rho = s^{n-si}\underline{0} \cdot \pi$, et le processus considéré se réduit en $t \star s^{n-i}\underline{0} \cdot \pi$ qui est dans $\perp\!\!\!\perp$ par l'hypothèse faite sur t. □

Rappelons que l'on peut faire apparaître des ensembles de temres dans les formules, à condition que ceux-ci soient à droite d'une flèche (Cf pages 11 et 14). On peut alors reformuler ce théorème de la manière suivante, la preuve étant rigoureusement la même :

$$T \in \bigcap_{n\in\mathbb{N}} |\forall X[(\{s^n\underline{0}\} \to X), Ent(s^n0) \to X]|.$$

On en déduit le résultat suivant, qui montre que le terme T effectue l'opération attendue sur les termes réalisant $Ent(s^n0)$, et peut être considéré comme une spécification de cette formule.

Théorème 2.2.29.
Soit $\nu \Vdash Ent(s^n0)$. Pour tout pile π et tout terme κ on a :

$$(T)\kappa \star \nu \cdot \pi \succ \kappa \star s^n\underline{0} \cdot \pi$$

Démonstration. Soit $\perp\!\!\!\perp = \{p \in \Lambda_c \star \Pi \ ; \ p \succ \kappa \star s^n\underline{0} \cdot \pi\}$. Soit $\Phi = \{\pi\}$. Le théorème 2.2.28 assure $T \Vdash (\{s^n\underline{0}\} \to \Phi), Ent(s^n0) \to \Phi$, et par hypothèse on a $\nu \Vdash Ent(s^n0)$. Comme $(T)\kappa \star \nu \cdot \pi \succ T \star \kappa \cdot \nu \cdot \pi$, il nous suffit pour avoir le résultat de démontrer que $\kappa \Vdash \{s^n\underline{0}\} \to \Phi$ pour ce $\perp\!\!\!\perp$, ce qui est trivial. □

Les fonctions récursives

L'opérateur de mise en mémoire permet également d'énoncer clairement des résultats concernant les fonctions récursives totales. Les résultats suivants expriment l'adéquation du calcul sur le type entier dans le cadre de la réalisabilité classique.

Théorème 2.2.30.
Soit f une fonction récursive totale, et t un λ-terme clos qui calcule celle-ci ; c'est-à-dire tel que si $\nu \simeq_\beta \underline{n}$, la réduction de tête faible de $(t)\nu$ mène à $\underline{m}$, où $m = f(n)$. Alors $(T)\lambda x(t)x \Vdash \forall x[Ent(x) \to Ent(fx)]$.

Démonstration. On se reportera à [18]. □

On peut aisément donner la spécification des formules de cette forme.

Théorème 2.2.31.
Soit $f : \mathbb{N} \to \mathbb{N}$, $\kappa \in \Lambda_c$ et $\nu \Vdash Ent(s^n 0)$. Alors si θ est un terme réalisant la formule $\forall x[Ent(x) \to Ent(fx)]$, on a quelle que soit la pile π :

$$(T)\kappa \star (\theta)\nu \cdot \pi \succ \kappa \star s^m \underline{0} \cdot \pi, \text{ où } m = f(n).$$

On peut alors se donner une instruction κ qui ne dispose d'aucune règle d'exécution, c'est-à-dire que l'exécution s'arrête lorsque celle-ci arrive en tête. Une telle instruction sera appelée *instruction stop*. Ce résultat assure que l'exécution du processus $(T)\kappa \star (\theta)\nu \cdot \pi$ se termine avec en tête de pile l'entier (calculé) $f(n)$.

Démonstration. Il suffit d'appliquer le théorème 2.2.29, comme $(\theta)\nu \Vdash Ent(f(s^n 0))$. □

Notons enfin que cette spécification ne suppose rien sur la fonction f, ce qui donne le corollaire suivant.

Corollaire 2.2.32.
Si la formule $\forall x[Ent(x) \to Ent(f(x))]$ est réalisé par une quasi-preuve indépendamment de $\perp\!\!\!\perp$, alors le symbole f désigne une fonction récursive totale.

Démonstration. Si cette formule est réalisée, le théorème précédent donne une manière de calculer la valeur de la fonction f en chaque entier. □

Ce résultat indique que les fonctions récursives totales sont les seules dont l'interprétation dans un modèle de la réalisabilité laisse stable l'ensemble des entiers du modèle.

Les formules équationnelles restreintes aux entiers

Considérons donc une formule équationnelle prouvable dans l'arithmétique de Peano du second ordre. Pour pallier l'absence de l'axiome de récurrence dans nos modèles, on peut restreindre tous les quantificateurs à la formule Ent. Notons déjà que si cette formule est de la forme $\forall x_1 \dots \forall x_k \; f(x_1, \dots, x_k) = 0$, elle est réalisée par I, et la spécification de sa forme restreinte aux entiers reste triviale. Le chapitre suivant présente une méthode permettant de donner un contenu opérationnel non-trivial à ce type de formule.

On présente ici deux résultats obtenus dans [18], concernant des théorèmes de la forme $\forall x \exists y \; f(x,y) = 0$ et $\exists x \forall y \; f(x,y) = 0$. On suppose donc disposer de deux preuves en arithmétique du second ordre sans axiome de récurrence : l'une de la formule $\forall x[Ent(x) \to \exists y(Ent(y), f(x,y) = 0)]$, et l'autre de la formule $\exists x[Ent(x), \forall y(Ent(y) \to f(x,y) = 0)]$.

Théorème 2.2.33.
Si $\vdash \theta : \forall x[Ent(x), \forall y(Ent(y) \to f(x,y) \neq 0) \to \bot]$, *alors quel que soit l'entier* n, *le terme* κ *et la pile* π, *on a :*

$$\theta \star \underline{n} \cdot (T)\kappa \cdot \pi \succ \kappa \star s^m \underline{0} \cdot \pi \text{ avec } f(n,m) = 0$$

où T *désigne toujours l'opérateur de mise en mémoire défini en 2.2.*

Remarques :
- Afin d'obtenir la spécification la plus claire possible, on n'a pas spécifié la formule $\forall x\{Ent(x) \to \forall X[(\forall y(Ent(y) \to f(x,y) = 0) \to X) \to X]\}$ mais une formule qui lui est équivalente en logique classique.
- L'hypothèse nécessaire à la démonstration de ce théorème est que θ réalise $\forall x[Ent(x), \forall y(Ent(y) \to f(x,y) \neq 0) \to \bot]$ quel que soit l'ensemble saturé $\perp\!\!\!\perp$.
- On peut encore prendre pour κ une instruction stop, ce qui fournit un programme permettant de calculer un entier m tel que $f(n,m) = 0$.
- Rappelons encore que ce résultat est indépendant des instructions disponibles, on a donc la même spécification quels que soient les axiomes que l'on utilise pour démontrer ce théorème.

On considère maintenant un théorème arithmétique de la forme $\exists x[Ent(x), \forall y(Ent(y) \to f(x,y) = 0)]$. On va définir un *jeu* associé à ce théorème. Les deux joueurs seront appelés $\exists$ et $\forall$ (respectivement pour Héloïse et Abélard). Intuitivement, le joueur $\forall$ attaque le théorème considéré alors que la joueuse $\exists$ le défend (elle cherche en fait à le prouver). Une partie se déroule comme suit :

1. $\exists$ joue un entier n, puis $\forall$ joue un entier p.
2. Si $f(n,p) = 0$ alors $\exists$ l'emporte ; sinon les joueurs choisissent chacun un nouvel entier.

$\forall$ l'emporte donc si et seulement si la partie ne s'arrête pas.
Il est clair que la joueuse $\exists$ possède une stratégie gagnante si et seulement si la formule considérée est vraie dans $\mathbb{N}$: jouer successivement 0, 1, 2,...

Le théorème 2.2.34 établit qu'un terme réalisant cette formule se comporte comme une implémentation à la fois de ce jeu et d'une stratégie gagnante pour la joueuse $\exists$.
On doit d'abord introduire de nouvelles instructions. On considère déjà une instruction notée κ qui possède la règle d'exécution suivante, quels que soient l'entier n, le terme ξ et la pile π :

$$\kappa \star s^n\underline{0} \cdot \xi \cdot \pi \succ \xi \star s^p\underline{0} \cdot \kappa_{np} \cdot \pi,$$

où κ_{np} est une suite double d'instructions « stop » et p un entier quelconque. On entend par « entier quelconque » que cette règle d'exécution est **non déterministe**, c'est-à-dire que cet entier n'est pas une fonction de (ξ, n, π). Intuitivement il s'agit d'une instruction interactive, et l'entier p est choisi par le joueur $\forall$ (à chaque fois que cette instruction arrive en tête) ; l'entier n aura été lui choisi au préalable, c'est-à-dire au cours de l'exécution du processus considéré : c'est donc le processus qui joue le rôle de la joueuse $\exists$ en choisissant ces entiers. L'instruction κ_{np} garde elle la trace des entiers qui ont été joués.

Théorème 2.2.34.
Si $\vdash \theta : \exists x[Ent(x), \forall y(Ent(y) \to f(x,y) = 0)]$, *alors* ***toute*** *réduction du processus* $\theta \star (T)\kappa \cdot \pi$ *se termine sur un processus de la forme* $\kappa_{np} \star \pi'$, *avec* $f(n,p) = 0$.

L'exécution de κ étant non-déterministe, dire que *toute* réduction du processus termine en étant indexée par un couple d'entiers (n,p) tel que $f(n,p) = 0$ signifie que la stratégie employée par la quasi-preuve θ est gagnante, c'est-à-dire que la partie se termine quel que soit le jeu de l'attaquant. Bien entendu, rien ne permet d'affirmer que l'entier n obtenu au terme d'une exécution donnée est l'un de ceux dont le théorème atteste l'existence.
Le terme θ est donc un programme qui, à chaque réplique p de l'attaquant, fournit un couple (n, ξ), n pouvant être considéré comme la « solution provisoire » et ξ comme « un gestionnaire d'exception » qui sera par exemple utilisé en cas de réponse pertinente de l'attaquant.

Remarque : On peut en fait associer un jeu à chaque formule du calcul des prédicats, et démontrer que toute quasi-preuve réalisant celle-ci implémente une partie dans laquelle la joueuse $\exists$ possède une stratégie gagnante (Cf le chapitre 5).

2.2.7 Définition récursive de prédicats

On se place pour cette section dans le cadre d'une réalisabilité effectuée à partir d'un ensemble $\mathscr{N}$ quelconque.

On rappelle ici le théorème du point fixe de Knaster-Tarski, et l'on montre comment utiliser celui-ci pour définir des prédicats.

Définition 2.2.35.
Un treillis complet est un ensemble ordonné $(E, \leq)$ tel que toute partie X de E possède une borne supérieure et une borne inférieure.

Définition 2.2.36.
Soient E et F des ensembles. On note $\ll$ l'ordre défini sur $\mathscr{P}(F)^E$ par :

$$f \ll g \text{ si et seulement si } \forall e \in E, f(e) \subseteq g(e)$$

Théorème 2.2.37.
Soient E et F des ensembles. Alors $(\mathscr{P}(F)^E, \ll)$ est un treillis complet.

Démonstration. Si $X \subset \mathscr{P}(F)^E$, on pose $S : e \mapsto \bigcup_{f \in X} f(e)$ et $I : e \mapsto \bigcap_{f \in X} f(e)$. On vérifie alors aisément que $\mathrm{Sup}(X) = S$ et $\mathrm{Inf}(X) = I$. □

Théorème 2.2.38 (Knaster-Tarski)**.**
Soit $(E, \leq)$ un treillis complet, et $f : (E, \leq) \to (E, \leq)$ une application monotone. Alors l'ensemble des points fixes de f n'est pas vide, et il contient ses bornes supérieure et inférieure.

Démonstration. On se reportera à [1]. □

Définition 2.2.39.
Soit $\Phi(y_1, \ldots, y_k, Y_1, \ldots, Y_m)$ une formule à paramètres dont les seules variables libres sont $y_1, \ldots, y_k$ et $Y_1, \ldots, Y_m$. On considère des prédicats $R_1, \ldots, R_m$ tels que R_i possède la même arité que le symbole Y_i.
On note alors $\phi(R_1, \ldots, R_m)$ le prédicat k-aire défini par

$$\phi(R_1, \ldots, R_m)(n_1, \ldots, n_k) = ||\Phi(n_1, \ldots, n_k, R_1, \ldots, R_m)||.$$

On a le résultat suivant, qui servira à plusieurs reprises par la suite.

Théorème 2.2.40.
Soit $\Phi(x_1, \ldots, x_k, X)$ une formule dont les seules variables libres sont $x_1, \ldots, x_k$ et X, cette dernière étant une variable de prédicat k-aire. Si X ne possède que des occurrences positives (resp. négatives) dans Φ, alors il existe $R \in \mathscr{P}(\Pi)^{\mathscr{N}^k}$ tel que :

$$R(n_1, \ldots, n_k) = \phi(R)(n_1, \ldots, n_k) \text{ quels que soient } n_1, \ldots, n_k \in \mathscr{N}.$$

Démonstration. Le lemme suivant permet d'appliquer le théorème 2.2.38, puisque $\mathscr{P}(\Pi)^{\mathscr{N}^k}$ est un treillis complet (théorème 2.2.37). □

Lemme 2.2.41.
Soit $\Phi(x_1, \ldots, x_k, X)$ une formule contenant des paramètres, dont les seules variables libres sont $x_1, \ldots, x_k$ et X, cette dernière étant une variable de prédicat k-aire.
Si X ne possède que des occurrences positives (resp. négatives) dans F, l'application

$$\begin{array}{ccc} (\mathscr{P}(\Pi)^{\mathscr{N}^k}, \ll) & \to & (\mathscr{P}(\Pi)^{\mathscr{N}^k}, \ll) \\ R & \mapsto & \phi(R) \end{array}$$

est croissante (resp. décroissante).

Démonstration. On prouve les deux résultats simultanément, par induction sur Φ. Soient deux prédicats R et R' tels que $R \ll R'$.
Montrons alors $\phi(R) \ll \phi(R')$ si X ne possède que des occurrences positives dans Φ (la démonstration dans l'autre cas est en tout point analogue). Si Φ est atomique, la démonstration est triviale.
Si $\Phi = \forall x \Psi(x)$, on obtient en appliquant l'hypothèse d'induction à $\Psi(p)$ que $\psi(R, p) \ll \psi(R', p)$ pour tout $p \in \mathscr{N}$, c'est-à-dire

$$||\Psi(n_1, \ldots, n_k, R, p)|| \subseteq ||\Psi(n_1, \ldots, n_k, R', p)||$$

quels que soient $n_1, \ldots, n_k$, $p \in \mathscr{N}$. On en déduit

$$\bigcup_{p \in \mathscr{N}} ||\Psi(n_1, \ldots, n_k, R, p)|| \subseteq \bigcup_{p \in \mathscr{N}} ||\Psi(n_1, \ldots, n_k, R', p)||,$$

c'est-à-dire $||\forall x \Psi(n_1, \ldots, n_k, R, x)|| \subseteq ||\forall x \Psi(n_1, \ldots, n_k, R', x)||$, ou encore $\phi(R) \ll \phi(R')$.
Si $\Phi = \forall Y \Psi(Y)$, où Y est d'arité q, on applique l'hypothèse de récurrence à la formule $\Psi(P)$ pour $P \in \mathscr{P}(\Pi)^{\mathscr{N}^q}$, ce qui assure

$$||\Psi(n_1, \ldots, n_k, R, P)|| \subseteq ||\Psi(n_1, \ldots, n_k, R', P)||$$

quels que soient $n_1, \ldots, n_k \in \mathscr{N}$. On en déduit

$$\bigcup_{P \in \mathscr{P}(\Pi)^{\mathscr{N}^q}} ||\Psi(n_1, \ldots, n_k, R, P)|| \subseteq \bigcup_{P \in \mathscr{P}(\Pi)^{\mathscr{N}^q}} ||\Psi(n_1, \ldots, n_k, R', P)||,$$

c'est-à-dire $||\forall Y \Psi(n_1, \ldots, n_k, R, Y)|| \subseteq ||\forall Y \Psi(n_1, \ldots, n_k, R', Y)||$, ou encore $\phi(R) \ll \phi(R')$.
Si $\Phi = \Psi_1 \to \Psi_2$, alors X ne possède par hypothèse que des occurrences positives dans Ψ_2 et que des occurrences négatives dans Ψ_1. L'hypothèse de récurrence appliquée à Ψ_2 puis Ψ_1 donne alors $\psi_2(R) \ll \psi_2(R')$ et $\psi_1(R') \ll \psi_1(R)$. On a donc quels que soient $n_1, \ldots, n_k \in \mathscr{N}$ que

$$||\Psi_2(n_1, \ldots, n_k, R)|| \subseteq ||\Psi_2(n_1, \ldots, n_k, R')||$$

et

$$||\Psi_1(n_1,\dots,n_k,R')|| \subseteq ||\Psi_1(n_1,\dots,n_k,R)||.$$

Le lemme 2.1.24 assure alors que

$$||\Psi_1(n_1,\dots,n_k,R) \to \Psi_2(n_1,\dots,n_k,R)|| \subseteq ||\Psi_1(n_1,\dots,n_k,R') \to \Psi_2(n_1,\dots,n_k,R')||$$

quels que soient $n_1,\dots,n_k \in \mathscr{N}$, c'est-à-dire que $\phi(R) \ll \phi(R')$.

□

2.2.8 Réalisabilité et présentation en séquents

La plupart des quasi-preuves données dans cette thèse ne sont pas directement obtenues par le lemme d'adéquation : afin de simplifier les termes obtenus, on exploite fréquemment des inclusions entre les valeurs de vérité de différentes formules (voir par exemple le lemme 2.1.20). On peut néanmoins souhaiter présenter ces preuves en séquents, afin d'illustrer la dialectique entre une démonstration et le comportement de la quasi-preuve associée. On est donc amené à ajouter de nouvelles règles de déduction permettant d'exploiter de telles égalités, puis à étendre le lemme d'adéquation.

Séquents : On considère des séquents de la forme

$$t_1 : A_1,\dots,t_n : A_n \vdash t : A$$

où :
- $t_1,\dots,t_n$ et t sont des termes, pas nécessairement clos.
- $A_1,\dots,A_n$ et A sont des formules usuelles.

Une expression de la forme $t_1 : A_1,\dots,t_n : A_n$ sera appelée *un contexte.* On s'autorise à commuter entre eux les différents éléments d'un contexte.

Règles de dérivation : On se donne pour commencer un certain nombre de règles, toutes dites *de remplacement*, qui permettent d'utiliser les inclusions éventuelles entre les valeurs de vérité de différentes formules. Chacune d'elle sera introduite au moment de son utilisation. Elles possèdent toutes la forme suivante :

$$\frac{t_1 : A_1,\dots,t_n : A_n \vdash u : A}{t_1 : A_1,\dots,t_n : A_n \vdash u : A'}\ (\mathrm{R})$$

Supposons que $x_1,\dots,x_n$, $X_1,\dots,X_p$ désignent les variables libres dans l'une ou l'autre des formules A et A'. On ne s'autorise à créer une telle règle que si quel que soit l'ensemble $\perp\!\!\!\perp$ saturé, les individus $a_1,\dots,a_n$ dans $\mathscr{N}$ ainsi que les prédicats $\Phi_1,\dots,\Phi_p$ d'arité convenable, on a

$$||A'[(a_i/x_i)_i,(\Phi_i/Y_i)_i]|| \subseteq ||A[(a_i/x_i)_i,(\Phi_i/Y_i)_i]||$$

On rajoute encore la règle suivante qui exprime que tout terme réalise $\top$:

$$\frac{}{\vdash t : \top}\ (\text{RéaAx}) \qquad \textit{Quel que soit le terme } t.$$

Enfin, on reprend les règles de typage donnée en 2.1.16 :

$$\frac{}{t_1 : A_1, \ldots, t_n : A_n \vdash t_j : A_j}\ (\text{Ax}) \qquad \textit{Pour tout } j \textit{ entre } 1 \textit{ et } n.$$

$$\frac{\Gamma \vdash t : A \to B \qquad \Delta \vdash u : A}{\Gamma, \Delta \vdash (t)u : B}\ (\to \text{élim})$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x\ t : A \to B}\ (\to \text{intro})$$

$$\frac{\Gamma \vdash t : (A \to B) \to A}{\Gamma \vdash (\mathsf{cc})t : A}\ (\text{Abs})$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall x A}\ (\forall\text{intro1}) \qquad \textit{Si } x \textit{ n'est libre pas dans } \Gamma.$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall X A}\ (\forall\text{intro2}) \qquad \textit{Si } X \textit{ n'est libre pas dans } \Gamma.$$

$$\frac{\Gamma \vdash t : \forall x A}{\Gamma \vdash t : A[\tau/x]}\ (\forall\text{élim1}) \qquad \textit{Pour tout terme } \tau \textit{ du langage.}$$

$$\frac{\Gamma \vdash t : \forall X A}{\Gamma \vdash t : A[F/X x_1 \ldots x_n]}\ (\forall\text{élim2}) \qquad \textit{Pour toute formule } F \textit{ avec au plus } n \textit{ variables libres.}$$

On peut alors démontrer que le lemme d'adéquation se prolonge aux séquents dérivables grâce à des règles de remplacement.

Théorème 2.2.42.
Soit F une formule close et t un terme clos tels que le séquent $\vdash t : F$ soit dérivable à l'aide des règles données ci-dessus. Alors $t \Vdash F$ quel que soit l'ensemble $\perp\!\!\!\perp$ saturé.

Le résultat découle du lemme suivant, adapté à une preuve par induction.

Lemme 2.2.43.
Soient $A_1, \ldots, A_n, A$ des formules dont les variables libres sont parmi $y_1, \ldots, y_m$, $Y_1, \ldots Y_l$. Notons k_i l'arité de la variable de prédicat Y_i.
Supposons que le séquent $x_1 : A_1, \ldots, x_n : A_n \vdash t : A$ soit dérivable avec les règles ci-dessus.
Soient $a_1, \ldots, a_m$ des éléments de $\mathscr{N}$, $\Phi_1, \ldots, \Phi_l$ des prédicats tels que Φ_i soit d'arité k_i.
Si l'on considère des termes $t_1, \ldots, t_n$ tels que $t_j \Vdash A_j[(a_i/y_i)_i, (\Phi_i/Y_i)_i]$, alors $t[t_1/x_1, \ldots, t_n/x_n] \Vdash A[(a_i/y_i)_i, (\Phi_i/Y_i)_i]$.

Démonstration. On procède par induction sur la longueur de la preuve. On considère la dernière règle utilisée.
Si c'est une règle usuelle de typage, on reprend la démonstration du lemme d'adéquation.
S'il s'agit de la règle assurant que tout terme réalise $\top$, le résultat est trivial.
Si la dernière règle appliquée est une règle de remplacement, on a affaire à une situation de la forme

$$\frac{x_1 : A_1, \ldots, x_n : A_n \vdash t : C}{x_1 : A_1, \ldots, x_n : A_n \vdash t : A}$$

Fixons des éléments $a_1, \ldots, a_k, \Phi_1, \ldots, \Phi_l$, et prenons des termes $t_1, \ldots, t_p$ tels que $x_j \Vdash A_j[(a_i/y_i)_i(\Phi_i/Y_i)_i]$, et une pile π dans $||A[(a_i/y_i)_i, (\Phi_i/Y_i)_i]||$. On doit montrer $t[t_1/x_1, \ldots, t_n/x_n] \star \pi \in \perp\!\!\!\perp$. Mais le fait d'employer cette règle assure que l'on a

$$||A[(a_i/y_i)_i, (\Phi_i/Y_i)_i]|| \subseteq ||C[(a_i/y_i)_i, (\Phi_i/Y_i)_i]||,$$

et donc $\pi \in ||C[(a_i/y_i)_i, (\Phi_i/Y_i)_i]||$: l'hypothèse de récurrence appliquée au séquent $x_1 : A_1, \ldots, x_n : A_n \vdash t : C$ donne alors le résultat. □

On peut déjà donner le résultat suivant, qui est celui utilisé dans la plupart des cas où l'on a recours à la règle de remplacement.

Lemme 2.2.44.
Soit F une formule dont les variables libres sont parmi $x, y_1, \ldots, y_p$. Alors quels que soient les entiers $m, n_1, \ldots, n_p$ on a :

$$\begin{aligned} ||\forall x(F(x, n_1, \ldots, n_p) \to x \neq m)|| &= ||F(m, n_1, \ldots, n_p) \to m \neq m|| \\ &= ||F(m, n_1, \ldots, n_p) \to \perp||. \end{aligned}$$

Démonstration. Si $k \neq m$ on a

$$||F(k, n_1, \ldots, n_p) \to k \neq m|| = ||F(k, n_1, \ldots, n_p) \to \top|| = \top.$$

Mais par contre,

$$||F(m, n_1, \ldots, n_p) \to m \neq m|| = ||F(m, n_1, \ldots, n_p) \to \perp||.$$

Cela donne le résultat considérant

$$||\forall x(F(x, n_1, \ldots, n_p) \to x \neq m|| = \bigcup_{k \in \mathcal{N}} ||F(k, n_1, \ldots, n_p) \to k \neq m||.$$

□

On rajoute donc, pour chaque formule F, la règle suivante :

$$\frac{\Gamma \vdash t : F(y) \to \bot}{\Gamma \vdash t : \forall x(F(x) \to x \neq y)}$$

CHAPITRE 3

Les entiers comme classes d'équivalences

On considère le modèle standard de l'arithmétique $\mathbb{N}$, à partir duquel on développe la théorie de la réalisabilité classique.

3.1 Une relation de préordre

3.1.1 Définitions

Le résultat suivant montre qu'à chaque individu x d'un modèle générique $\mathscr{M}$ est associé un unique entier du modèle : on peut donc considérer chaque entier comme une classe d'équivalence, celle des individus qui possèdent cet entier pour rang. Cela donne une manière alternative de raisonner dans les modèles de la réalisabilité, et de donner un contenu opérationnel aux formules de l'arithmétique : au lieu de restreindre ces formules aux entiers, on pourra considérer le modèle quotienté et se restreindre aux prédicats saturés pour cette relation d'équivalence. On obtient deux modèles isomorphes de l'arithmétique du second ordre. Ce chapitre étudie la seconde approche.

Notation : On dénotera par $y \simeq 0$ la formule $\forall y'(sy' \neq y)$.

Théorème 3.1.1.
La formule

$$\forall x \exists! n \exists! y \{Ent(n), y \simeq 0, x = n + y\}$$

est réalisée par une quasi-preuve.

Démonstration. La formule $\forall x \exists n \exists y (Ent(n), y \simeq 0, x = n + y)$ est réalisée, puisqu'elle est déductible en logique classique du second ordre de la formule
$\forall X\{\forall x[\forall y(sy = x \to X(y)) \to X(x)] \to \forall z X(z)\}$.
Pour prouver l'unicité, il suffit de montrer celle-ci en logique classique du

second ordre en partant de formules elles-mêmes réalisables par des quasi-preuves. Supposons donc avoir deux couples (y,n) et (y',n') convenant. On a alors $y+n=y'+n'$. Si $n=n'$, on en déduit immédiatement $y=y'$ puisque la formule $\forall x\forall x'\forall z(x+z=x'+z\to x=x')$ est réalisée par $\lambda x\ x$. Si par exemple $n<n'$, on en déduit de même $y=y'+(n'-n)$; d'où $y=s(y'+(n'-n-1))$, ce qui contredit l'hypothèse $y\simeq 0$. □

Définition 3.1.2.
Pour tout modèle générique $\mathscr{M}$ et tout élément x de $\mathscr{M}$, on appelle « rang de x » et l'on note $rg(x)$ l'unique élément n de $\mathscr{M}$ tel que $\mathscr{M}$ satisfait les formules $Ent(n)$ et $\exists y'(y'\simeq 0\wedge x=y'+n)$.

Commençons par étendre l'ordre naturel sur les entiers du modèle en une relation bien fondée sur $\mathscr{M}$. Se pose déjà le problème de savoir comment définir dans notre langage une telle relation.

Remarquons déjà que si f est la fonction définie par

$$f(x,y)=\begin{cases}1 \text{ si } x\geqslant y\\ 0 \text{ sinon}\end{cases}$$

l'équation $f(x,y)=1$ ne définit pas la relation cherchée. Les deux résultats suivants indiquent en effet que, via un symbole de fonction de notre langage, on ne peut pas définir une relation binaire qui soit totale mais pas symétrique. Etant donné un modèle de la réalisabilité, cela donne un premier exemple d'application définie sur celui-ci qui n'est pas représentée par un symbole de fonction de notre langage : celle qui à un individu du modèle associe son rang.

Théorème 3.1.3.
Soit $f:\mathbb{N}^2\to\mathbb{N}$ telle qu'il existe des entiers n et p tels que $f(n,p)=0$ et $f(p,n)\neq 0$. Alors la formule $\forall x\forall y(f(x,y)=0\vee f(y,x)=0)$ n'est pas réalisable indépendamment de $\perp\!\!\!\perp$.

On a d'abord besoin du

Lemme 3.1.4.
On a les résultats suivants :

1. *I réalise $\forall X\forall Y(X\vee Y\to\{(X\to\bot),(Y\to\bot)\to\bot\})$.*
2. *$\lambda x\lambda y\lambda z(\mathsf{cc})\lambda k(x)(k)y(k)z$ réalise la formule suivante*

$$\forall X\forall Y(\{(X\to\bot),(Y\to\bot)\to\bot\}\to X\vee Y\}).$$

Démonstration. 1. Il suffit de remarquer que, quelles que soient les parties X et Y de Π, la valeur de vérité de $\forall Z\{(X\to Z),(Y\to Z)\to Z\}$ contient celle de $(X\to\bot),(Y\to\bot)\to\bot$.

2. Soient X, Y et Z inclus dans Π, $\pi \in Z$, trois termes u, v et w réalisant respectivement $\{(X \to \bot), (Y \to \bot) \to \bot\}$, $(X \to Z)$ et $(Y \to Z)$. Nous devons montrer que le processus $\lambda x \lambda y \lambda z (\mathsf{cc}) \lambda k (x)(k) y (k) z \star u \cdot v \cdot w \cdot \pi$ est dans $\perp\!\!\!\perp$. Il nous suffit donc de montrer que $u \star (\mathsf{k}_\pi) u \cdot (\mathsf{k}_\pi) v \cdot \pi$ est dans $\perp\!\!\!\perp$. Il suffit pour cela de vérifier $(\mathsf{k}_\pi) u \Vdash X \to \bot$ et $(\mathsf{k}_\pi) v \Vdash Y \to \bot$. Mais si l'on prend un terme t réalisant X et une pile ρ, le processus $(\mathsf{k}_\pi) u \star t \cdot \rho$ se réduit en $u \star t \cdot \pi$ qui est dans $\perp\!\!\!\perp$ par l'hypothèse faite sur u. On raisonne de même pour montrer $(\mathsf{k}_\pi) v \Vdash Y \to \bot$.

□

Démonstration du théorème 3.1.3. Considérant le lemme précédent et le fait que l'on peut réaliser l'équivalence $\forall x \forall y ((x = y \to \bot) \leftrightarrow x \neq y)$, il suffit de démontrer qu'il n'existe pas de quasi-preuve ξ réalisant la formule $\forall x \forall y (f(x,y) \neq 0, f(y,x) \neq 0 \to \bot)$. Mais on a :

$$|f(n,p) \neq 0, f(p,n) \neq 0 \to \bot| = |\bot, \top \to \bot|$$

$$\text{et } |f(p,n) \neq 0, f(n,p) \neq 0 \to \bot| = |\top, \bot \to \bot|.$$

Le résultat découle donc du théorème 2.2.20. □

Théorème 3.1.5.
Soit $f : \mathbb{N}^2 \to \mathbb{N}$ telle qu'il existe n et p tels que $f(n,p) = 0$ et $f(p,n) \neq 0$. Alors la formule $\forall x \forall y (f(x,y) \neq 0 \vee f(y,x) \neq 0)$ n'est pas réalisable pour tout $\perp\!\!\!\perp$.

Démonstration. On effectue un raisonnement similaire à celui effectué ci-dessus. □

On est donc contraint de définir cette relation par un prédicat bien choisi, que l'on notera $\sqsubseteq$.

Définition 3.1.6.
On note $\sqsubseteq$ l'unique élément de $\mathscr{P}(\Pi)^{\mathbb{N}^2}$ vérifiant pour tous les entiers n, p la relation

$$n \sqsubseteq p \;=\; \left|\left| \forall n' \left(\forall p' \left(n' \sqsubseteq p' \to s p' \neq p \right) \to s n' \neq n \right) \right|\right|$$

Cet élément est défini grâce au théorème 2.2.40 appliqué à la formule $\Phi(R)$ suivante :

$$\forall n' \left(\forall p' \left(R(n',p') \to s p' \neq p \right) \to s n' \neq n \right).$$

Remarquons déjà que l'on peut aisément expliciter ce prédicat, ce qui assure par ailleurs l'unicité du point fixe considéré.

Théorème 3.1.7.
On a les égalités suivantes, quels que soient les entiers n et p.
- $0 \sqsubseteq n = \top$
- $sn \sqsubseteq 0 = ||\top \to \bot||$
- $sn \sqsubseteq sp = ||(n \sqsubseteq p \to \bot) \to \bot||$

Démonstration. Il suffit d'utiliser le lemme 2.2.44 qui assure que $||\forall x(F(x) \to x \neq p)|| = ||F(p) \to \bot||$.

□

C'est-à-dire que quels que soient les entiers n et p, la formule $n \sqsubseteq p$ correspond à une suite de négation de longueur paire qui mène soit au type $\top$, soit au type $\top \to \bot$.

Notations : On notera, pour toute formule F, par $\neg^2 F$ la formule $((F \to \bot) \to \bot)$. Et récursivement, la formule $\neg^{2(n+1)} F$ sera une notation pour la formule $((\neg^{2n} F \to \bot) \to \bot)$.

Avec ces notations, on déduit du théorème précédent que quels que soient les entiers n et p :

$$n \sqsubseteq p = \begin{cases} ||\neg^{2n}\top|| & \text{si } n \leqslant p \\ ||\neg^{2p}(\top \to \bot)|| & \text{si } n > p \end{cases}$$

Définition 3.1.8.
Soit $\mathscr{M}$ un modèle générique. On appelle pseudo-zéro tout élément y de $\mathscr{M}$ tel que $\mathscr{M} \vDash \forall y'(sy' \neq y)$.

On vérifie facilement que dans un modèle générique, la formule $x \sqsubseteq y$ est vraie si et seulement si x et y sont deux individus tels que le rang de x soit au plus le rang de y.

Lemme 3.1.9.
Soit $\mathscr{M}$ un modèle générique, et x un élément de $\mathscr{M}$. Alors le rang de x est non nul si et seulement si il existe un élément x' de $\mathscr{M}$ tel que $\mathscr{M} \vDash sx' = x$. Et si un tel élément x' existe, il est unique et a pour rang $rg(x) - 1$.

Démonstration. Découle trivialement du théorème 3.1.1 ainsi que de l'injectivité de la fonction interprétant le symbole s dans $\mathscr{M}$, puisque I réalise $\forall x \forall y(x \neq y \to sx \neq sy)$. □

Théorème 3.1.10.
Soient x et y deux éléments d'un modèle générique $\mathscr{M}$, alors

$$\mathscr{M} \models x \sqsubseteq y \text{ ssi } rg(x) \leqslant rg(y)$$

Démonstration. On va démontrer le résultat par récurrence sur le rang de x.
Si $rg(x) = 0$: on doit montrer $\mathscr{M} \vDash x \sqsubseteq y$ pour tout élément y du modèle. Il est équivalent de montrer $\mathscr{M} \vDash \forall x'(\forall y'(x' \sqsubseteq y' \to sy' \neq y) \to sx' \neq x)$, puisque ces deux formules ont la même valeur de vérité (et donc on réalise leur équivalence via $\lambda x\ x$). Or on a par le lemme précédent que pour tout élément x' du modèle, la formule $sx' = x$ est réfutée dans le modèle. C'est donc que $\mathscr{M} \vDash sx' \neq x$ pour tout élément x'. Donc $\mathscr{M}$ satisfait la formule $x \sqsubseteq y$.
Si $rg(x) = n + 1$: Supposons dans un premier temps avoir un élément y du modèle tel que $\mathscr{M} \vDash x \sqsubseteq y$ et $rg(y) > rg(x)$. Soit x' le prédécesseur de x dans $\mathscr{M}$. Il vient $\mathscr{M} \vDash \forall y'(x' \sqsubseteq y' \to sy' \neq y) \to \bot$. C'est donc que
$\mathscr{M} \nvDash \forall y'(x' \sqsubseteq y' \to sy' \neq y)$, c'est-à-dire qu'il existe un élément y' tel que $\mathscr{M} \vDash x' \sqsubseteq y'$ et $\mathscr{M} \vDash sy' = y$. Par hypothèse de récurrence, on a
$rg(x') = rg(x) - 1 \leqslant rg(y') = rg(y) - 1$, ce qui est une contradiction.
Réciproquement, supposons avoir un élément y tel que $rg(y) \geqslant rg(x)$. On doit montrer $\mathscr{M} \vDash \forall x'(\forall y'(x' \sqsubseteq y' \to sy' \neq y) \to sx' \neq x)$. Soit x' tel que
$\mathscr{M} \vDash sx' = x$. Il suffit de montrer $\mathscr{M} \nvDash \forall y'(x' \sqsubseteq y' \to sy' \neq y)$. Soit alors y' tel que $\mathscr{M} \vDash sy' = y$; il suffit de constater que $\mathscr{M} \vDash x' \sqsubseteq y'$ est vrai par hypothèse de récurrence. □

3.1.2 Transitivité

On commence par s'intéresser au théorème exprimant la transitivité de la relation binaire induite par le prédicat $\sqsubseteq$ dans un modèle générique. Donnons déjà un réalisateur de cette formule.

Théorème 3.1.11.
La formule $\forall x \forall y \forall z(x \sqsubseteq y,\ y \sqsubseteq z \to x \sqsubseteq z)$ est réalisé par la quasi-preuve suivante :

$$(Y)\lambda r \lambda u \lambda v \lambda w(u)\lambda x(v)\lambda y(w)(r)xy.$$

On donne deux démonstrations de ce résultat : la première présentée en séquents, qui souligne l'aspect logique du résultat en donnant la preuve dont est issu ce terme, et la seconde dans le formalisme de la réalisabilité, qui a le mérite d'être plus concise.

Première démonstration. On commence par considérer les deux règles de remplacement (Cf la section 2.2) suivantes :

$$\frac{\Gamma \vdash t : \forall x'(\forall y'(x' \sqsubseteq y' \to sy' \neq y) \to sx' \neq x)}{\Gamma \vdash t : x \sqsubseteq y} \qquad \frac{\Gamma \vdash t : x \sqsubseteq y}{\Gamma \vdash t : \forall x'(\forall y'(x' \sqsubseteq y' \to sy' \neq y) \to sx' \neq x)}$$

Soit alors $F(a)$ la formule suivante :

$$\forall y \forall z(a \sqsubseteq y, y \sqsubseteq z \to a \sqsubseteq z)$$

Considérant le théorème 2.2.18, il suffit de démontrer que le terme $\lambda r\lambda u\lambda v\lambda w(u)\lambda x(v)\lambda y(w)(r)xy$ réalise les formules $\forall a(F(a) \to F(sa))$ et $\top \to F(0)$.

Montrons déjà le premier point. On prend les notations suivantes :

- R pour le contexte $r : F(a)$,
- X pour le contexte $x : a \sqsubseteq b'$,
- Y pour le contexte $y : b' \sqsubseteq c'$,
- W pour le contexte $w : \forall \bar{c}(a \sqsubseteq \bar{c} \to s\bar{c} \neq c)$,
- V pour le contexte $v : b \sqsubseteq c$,
- U pour le contexte $u : sa \sqsubseteq b$.

On considère la dérivation suivante, qui est « découpée en morceaux » pour des raisons évidentes de mise en page.

$$\dfrac{\dfrac{R \vdash r : a \sqsubseteq b', b' \sqsubseteq c' \to a \sqsubseteq c' \qquad X \vdash x : a \sqsubseteq b'}{R, X \vdash (r)x : b' \sqsubseteq c' \to a \sqsubseteq c'} \qquad Y \vdash y : b' \sqsubseteq c'}{R, X, Y \vdash (r)xy : a \sqsubseteq c'}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{R, X, Y \vdash (r)xy : a \sqsubseteq c'} \qquad \dfrac{W \vdash w : \forall \bar{c}(a \sqsubseteq \bar{c} \to s\bar{c} \neq c)}{W \vdash w : a \sqsubseteq c' \to sc' \neq c}}{R, X, Y, W \vdash (w)(r)xy : sc' \neq c}}{R, X, W \vdash \lambda y(w)(r)xy : b' \sqsubseteq c' \to sc' \neq c}}{R, X, W \vdash \lambda y(w)(r)xy : \forall c'(b' \sqsubseteq c' \to sc' \neq c)}$$

Considérant par ailleurs la dérivation suivante

$$\dfrac{\dfrac{V \vdash v : b \sqsubseteq c}{V \vdash v : \forall b'(\forall c'(b' \sqsubseteq c' \to sc' \neq c) \to sb' \neq b)}\,\mathrm{R}}{V \vdash v : \forall c'(b' \sqsubseteq c' \to sc' \neq c) \to sb' \neq b}$$

On en déduit que le séquent $R, X, W, V \vdash (v)\lambda y(w)(r)xy : sb' \neq b$ est dérivable. On peut alors conclure comme suit :

$$\vdots$$
$$R,X,W,V \vdash (v)\lambda y(w)(r)xy : sb' \neq b$$
$$R,W,V \vdash \lambda x(v)\lambda y(w)(r)xy : a \sqsubseteq b' \to sb' \neq b$$
$$R,W,V \vdash \lambda x(v)\lambda y(w)(r)xy : \forall b'(a \sqsubseteq b' \to sb' \neq b)$$
$$\dfrac{R,W,V \vdash \lambda x(v)\lambda y(w)(r)xy : \forall b'(a \sqsubseteq b' \to sb' \neq b) \qquad \dfrac{U \vdash sa \sqsubseteq b}{U \vdash u : \forall b'(a \sqsubseteq b' \to sb' \neq b) \to sa \neq sa}}{R,W,V,U \vdash (u)\lambda x(v)\lambda y(w)(r)xy : sa \neq sa}$$
$$R,V,U \vdash \lambda w(u)\lambda x(v)\lambda y(w)(r)xy : \forall c'(a \sqsubseteq c' \to sc' \neq c) \to sa \neq sa$$
$$R,V,U \vdash \lambda w(u)\lambda x(v)\lambda y(w)(r)xy : \forall a'(\forall c'(a' \sqsubseteq c' \to sc' \neq c) \to sa' \neq sa) \quad \text{R}$$
$$R,V,U \vdash \lambda w(u)\lambda x(v)\lambda y(w)(r)xy : sa \sqsubseteq c \quad \text{R}$$
$$R,U \vdash \lambda v\lambda w(u)\lambda x(v)\lambda y(w)(r)xy : b \sqsubseteq c \to sa \sqsubseteq c$$
$$R \vdash \lambda u\lambda v\lambda w(u)\lambda x(v)\lambda y(w)(r)xy : sa \sqsubseteq b, b \sqsubseteq c \to sa \sqsubseteq c$$
$$R \vdash \lambda u\lambda v\lambda w(u)\lambda x(v)\lambda y(w)(r)xy : \forall c(sa \sqsubseteq b, b \sqsubseteq c \to sa \sqsubseteq c)$$
$$R \vdash \lambda u\lambda v\lambda w(u)\lambda x(v)\lambda y(w)(r)xy : F(sa)$$
$$\vdash \lambda r\lambda u\lambda v\lambda w(u)\lambda x(v)\lambda y(w)(r)xy : F(a) \to F(sa)$$
$$\vdash \lambda r\lambda u\lambda v\lambda w(u)\lambda x(v)\lambda y(w)(r)xy : \forall a(F(a) \to F(sa))$$

Pour le second point, on prend les notations suivantes :
- R pour le contexte $r : \top$,
- W pour le contexte $w : \forall \bar{c}(a' \sqsubseteq \bar{c} \to s\bar{c} \neq c)$,
- V pour le contexte $v : b \sqsubseteq c$,
- U pour le contexte $u : 0 \sqsubseteq b$.

La dérivation suivante donne alors le résultat.

$$R,W,V,U \vdash (u)\lambda x(v)\lambda y(w)(r)xy : \top$$
$$R,W,V,U \vdash (u)\lambda x(v)\lambda y(w)(r)xy : sa' \neq 0 \quad \text{R}$$
$$R,V,U \vdash \lambda w(u)\lambda x(v)\lambda y(w)(r)xy : \forall c'(a' \sqsubseteq c' \to sc' \neq c) \to sa' \neq 0$$
$$R,V,U \vdash \lambda w(u)\lambda x(v)\lambda y(w)(r)xy : \forall a'(\forall c'(a' \sqsubseteq c' \to sc' \neq c) \to sa' \neq 0) \quad \text{R}$$
$$R,V,U \vdash \lambda w(u)\lambda x(v)\lambda y(w)(r)xy : 0 \sqsubseteq c \quad \text{R}$$
$$R,U \vdash \lambda v\lambda w(u)\lambda x(v)\lambda y(w)(r)xy : b \sqsubseteq c \to 0 \sqsubseteq c$$
$$R \vdash \lambda u\lambda v\lambda w(u)\lambda x(v)\lambda y(w)(r)xy : 0 \sqsubseteq b, b \sqsubseteq c \to 0 \sqsubseteq c$$
$$R \vdash \lambda u\lambda v\lambda w(u)\lambda x(v)\lambda y(w)(r)xy : F(0)$$
$$\vdash \lambda r\lambda u\lambda v\lambda w(u)\lambda x(v)\lambda y(w)(r)xy : \top \to F(0)$$

□

Deuxième démonstration. On note θ le terme considéré. On va prouver par récurrence sur m que quels que soient les entiers n et p ainsi que la pile π dans $||m \sqsubseteq n,\ n \sqsubseteq p \to m \sqsubseteq p||$, le processus $\theta \star \pi$ est dans $\perp\!\!\!\perp$.
Si $m = 0$, le résultat est trivial car $||m \sqsubseteq n,\ n \sqsubseteq p \to m \sqsubseteq p|| = \top$.
Sinon, π est de la forme $u \cdot v \cdot w \cdot \rho$ avec $u \Vdash m \sqsubseteq n$, $v \Vdash n \sqsubseteq p$ et $w \Vdash \forall y(m-1 \sqsubseteq y \to sy \neq p)$. Le processus considéré se réduit alors en $u \star \lambda x(v)\lambda y(w)(\theta)xy \cdot \rho$. Puisque $u \Vdash \forall y(m-1 \sqsubseteq y \to sy \neq n) \to \perp$, il suffit de montrer $\lambda x(v)\lambda y(w)(\theta)xy \Vdash \forall y(m-1 \sqsubseteq y \to sy \neq n)$. Si $n = 0$, cela est trivial ;

sinon, on considère un terme u'réalisant $m-1 \sqsubseteq n-1$, et on est ramené à montrer $\lambda x(v)\lambda y(w)(\theta)xy \star u' \cdot \rho' \in \perp\!\!\!\perp$, pour toute pile ρ'. Or ce processus se réduit en $v \star \lambda y(w)(\theta)u'y \cdot \rho'$, et puisque v réalise $\forall y(n-1 \sqsubseteq y \to sy \neq p) \to \bot$, il nous suffit de démontrer que $\lambda y(w)(\theta)u'y$ réalise $\forall y(n-1 \sqsubseteq y \to sy \neq p)$. On peut encore supposer $p \neq 0$, et on s'est ramené à démontrer que si $v' \Vdash n-1 \sqsubseteq p-1$ et $\rho'' \in \Pi$, alors $\lambda y(w)(\theta)u'y \star v' \cdot \rho'' \in \perp\!\!\!\perp$. Mais ce processus se réduit en $w \star (\theta)u'v' \cdot \rho''$, et comme $w \Vdash m-1 \sqsubseteq\!\!\to p-1 \to \bot$, il suffit de prouver que $(\theta)u'v'$ réalise $m-1 \sqsubseteq p-1$, ce qui est vrai par hypothèse de récurrence. □

Soit θ la quasi-preuve donnée dans le théorème précédent. On considère trois termes u, v, w quelconques, une pile π et l'on observe la suite des réductions de $\theta \star u \cdot v \cdot w \cdot \pi$. Il se réduit en $(u) \star \lambda x(v)\lambda y(w)((Y)\theta)xy \cdot \pi$ La suite dépend évidemment du terme u, mais si l'on suppose que celui-ci finit au bout d'un certain nombre d'étapes de réduction par « rendre la main », c'est-à-dire par mettre son argument en tête et par lui donner un terme u', on obtient :

$$\theta \star u \cdot v \cdot w \cdot \pi \succ (v) \star \lambda y(w)((Y)\theta)u'y \cdot \pi$$

Encore une fois, si l'on suppose que v (resp. w) finit lui aussi par mettre son argument en tête et lui donner un terme v' (resp. w'), on obtient la suite de réductions suivantes :

$$\begin{aligned} \theta \star u \cdot v \cdot w \cdot \pi &\succ (u) \star \lambda x(v)\lambda y(w)((Y)\theta)xy \cdot \pi \\ &\succ (v) \star \lambda y(w)((Y)\theta)u'y \cdot \pi \\ &\succ w \star ((Y)\theta)u'v' \cdot \pi \\ &\succ \theta \star u' \cdot v' \cdot w' \cdot \pi \end{aligned}$$

Une première interprétation : On remarque que l'utilisation du combinateur de point fixe Y permet d'écrire un terme dont l'exécution est récursive. L'emploi de Y pour la construction d'un terme correspond donc à celui d'une « boucle while » dans l'écriture d'un programme. Alors que la réalisation de formules restreintes aux entiers correspond à la programmation itérative (utilisant des « boucles for »), la réalisation de formules dans le modèle quotienté (qui est un modèle élémentairement équivalent au premier modèle de l'arithmétique de Peano) correspond elle au paradigme de la programmation récursive. Il y a donc une correspondance entre paradigmes de raisonnement et paradigmes de programmation.

On peut se demander pourquoi cette stratégie permet de prouver le théorème. Le mécanisme qu'effectue un réalisateur de ce théorème doit permettre, étant donné trois suites de négations de longueur paire dont l'une au moins mène au type $\top \to \bot$, d'extraire un terme réalisant $\top \to \bot$.

En effet dire que θ réalise le théorème considéré implique que, quels que soient les entiers $n,\ m,\ p$ non nuls, θ réalise la formule :

$$\neg^{2\mathrm{Min}(n,m)} M(n,m),\ \neg^{2\mathrm{Min}(m,p)} M(m,p),\ \neg^{2\mathrm{Min}(n,p)-1} M(n,p) \to \bot,$$

où $M(a,b)$ dénote la formule $\top$ si $a \leqslant b$, et la formule $\top \to \bot$ si $a > b$. Le terme que nous avons trouvé va intuitivement remonter ces trois suites de négation l'une après l'autre, en retirant deux négations à chaque fois, et en suivant l'ordre dans lequel elles lui sont données. Il ne réussira donc à aboutir au type $\top \to \bot$ que si celui-ci se trouve dans la première des trois dont la taille est minimale. En effet, si θ met en tête au cours de son exécution (dans un contexte donné) un terme ne réalisant a priori que $\top$ (c'est-à-dire un terme quelconque), alors rien ne dit que celui-ci finira par mettre son argument en tête, c'est-à-dire par rendre la main à θ. Au contraire, un terme réalisant une formule de la forme $(F \to \bot) \to \bot$ finira toujours par mettre son argument en tête en lui donnant à son tour un argument. Et le fait que le type $\top \to \bot$ se trouve au bout de la suite de négation la plus courte des trois se prouve via le résultat élémentaire d'arithmétique suivant :

$$\forall i \forall j \forall k [\mathrm{Min}(i,j) < \mathrm{Min}(i,k) \to i > j]$$

Celui-ci permet d'assurer que :

1. si la première suite de négation est de longueur plus courte que la troisième, alors elle pointe sur $\top \to \bot$.
2. si la seconde suite de négation est de longueur strictement plus petite que la première, alors elle pointe sur $\top \to \bot$.
3. si la troisième suite de négation est de longueur strictement plus courte que la première et la seconde, alors elle pointe sur $\top \to \bot$.

Donc dans tous les cas la première suite de négation à être de taille minimale pointe sur le type $\top \to \bot$.

Spécification du théorème de transitivité

Commençons par énoncer la spécification de la formule $(\top \to \bot) \to \bot$.

Théorème 3.1.12.
Soit u réalisant la formule $(\top \to \bot) \to \bot$. Alors quel que soient le terme t et la pile π, on a

$$u \star t \cdot \pi \succ t \star \rho,$$

où ρ est une pile non vide.

La formule $(\top \to \bot) \to \bot$ spécifie donc les termes qui « utilisent leur argument », au sens où ils finissent par le faire apparaître en tête d'une pile non vide. On démontre aisément que la réciproque est vraie.

Démonstration du théorème 3.1.12. Soit $\perp\!\!\!\perp =< u \star t \cdot \pi >$. Considérant que u réalise $(\top \to \bot) \to \bot$ et que $u \star t \cdot \pi \notin \perp\!\!\!\perp$, on en déduit $t \nVdash \top \to \bot$. C'est-à-dire qu'il existe $\rho \in ||\top \to \bot||$ tel que $t \star \rho \notin \perp\!\!\!\perp$; autrement dit le processus $u \star t \cdot \pi$ se réduit en $t \star \rho$. Il reste à voir que $||\top \to \bot||$ est égal à $\{v \cdot \pi';\ v \in \Lambda_c, \pi' \in \Pi\}$ pour conclure. □

Pour énoncer clairement la spécification des formules de la forme $n \sqsubseteq n+p$, on se donne une suite d'instructions $(\gamma_i^1)_i$, possédant les règles d'exécutions suivantes :

$$\gamma_{i+1}^1 \star t \cdot \pi \succ t \star \gamma_i^1 \cdot \pi$$

En particulier l'instruction γ_0^1 ne possède pas de règle d'exécution, et si une instruction γ_{i+1}^1 arrive en tête d'une pile vide, l'exécution s'arrête.

Bien évidemment, il n'est pas nécessaire de considérer ces nouvelles instructions, puisque leur comportement peut être obtenu avec des λ-termes usuels. Cette manière de procéder permet cependant d'exprimer de manière plus claire les spécifications à venir. Par ailleurs, il est important de noter que l'on ne demande pas aux ensembles $\perp\!\!\!\perp$ considérés d'être saturés pour leur règle d'exécution : ces instructions ne possèdent aucun contenu axiomatique, et ne servent en aucun cas à écrire des quasi-preuves. Le lemme 2.2.2 assure néanmoins que le complémentaire d'un fil est saturé pour chaque instruction γ_i^1.

Lemme 3.1.13.
Soit p un processus, et $\perp\!\!\!\perp =< p >$. Alors γ_n^1 réalise la formule $\neg^{2n}\top$, quel que soit l'entier n.

Démonstration. On prouve le résultat par induction sur n.
Si $n = 0$, il n'y a rien à démontrer.
Si $n > 0$, on prend $t \in |\neg^{2(n-1)}\top \to \bot|$, $\pi \in \Pi$, et l'on doit montrer que le processus $\gamma_n^1 \star t \cdot \pi$ est dans $\perp\!\!\!\perp$. Celui-ci réduit en $t \star \gamma_{n-1}^1 \cdot \pi$, et puisque le lemme 2.2.2 assure que $\perp\!\!\!\perp =< p >$ est saturé pour l'instruction γ_n^1, il suffit de démontrer que ce dernier processus est dans $\perp\!\!\!\perp$. On déduit alors le résultat de l'hypothèse d'induction. □

Réciproquement, un terme réalisant la formule $\neg^{2n}\top$ a un comportement analogue à celui de l'instruction γ_n^1. Montrons déjà le lemme suivant, qui sera souvent utilisé dans la suite.

Lemme 3.1.14.
Soit p un processus, et $\perp\!\!\!\perp =< p >$. Si $\gamma_n^1 \nVdash \neg^{2n}\top \to \bot$, alors $\gamma_0^1 \nVdash \top \to \bot$.

Démonstration. Nous allons prouver la formule contraposée par récurrence sur n. Si $n = 0$ il n'y a rien à démontrer. Sinon, on considère un terme t réalisant $\neg^{2n}\top$ et une pile π. On doit montrer que le processus $\gamma_n^1 \star t \cdot \pi$ est dans $\perp\!\!\!\perp$. Celui-ci se réduit en $t \star \gamma_{n-1}^1 \cdot \pi$, et considérant le lemme 2.2.2, il suffit de démontrer que ce dernier processus est dans $\perp\!\!\!\perp$. Mais l'hypothèse de récurrence assure que $\gamma_{n-1}^1 \Vdash \neg^{2(n-1)}\top \to \bot$, ce qui termine la démonstration. □

Théorème 3.1.15.
Soit $n > 0$ et θ réalisant $n \sqsubseteq n+p$ pour tout $\perp\!\!\!\perp$. Alors pour $k \leqslant n-1$, l'exécution du processus $\theta \star \gamma_k^1 \cdot \pi$ se termine sur un processus de la forme $\gamma_0^1 \star t \cdot \rho$, quelle que soit la pile π.

Un tel résultat est évidemment inutile du point de vue de la programmation, puisque rien ne garantit que l'exécution s'arrête dans le cas où $k > n-1$.

Démonstration. Soit $\perp\!\!\!\perp =< \theta \star \gamma_{n-1}^1 \cdot \pi >$. Puisque θ réalise la formule $(n-1 \sqsubseteq n+p-1 \to \bot) \to \bot$ et que $\theta \star \gamma_{n-1}^1 \cdot \pi \notin \perp\!\!\!\perp$, on en déduit que γ_{n-1}^1 ne réalise pas $n-1 \sqsubseteq n+p-1 \to \bot$ pour ce $\perp\!\!\!\perp$. Le lemme 3.1.14 assure alors $\gamma_0^1 \nVdash \top \to \bot$, c'est-à-dire qu'il existe une pile ρ et un terme t tels que $\gamma_0^1 \star t \cdot \rho$ apparaisse dans le fil considéré, ce qui termine la démonstration. On raisonnerait de même avec $k < n-1$.

□

Remarque : La suite d'instructions γ permet d'énoncer la spécification des formules de la forme $\neg^{2n}\top$. On peut la voir comme étant un programme écrit spécialement pour vérifier le comportement d'un autre programme θ. En effet, γ_n^1 se contente de donner à θ ce qu'il lui faut pour poursuivre son exécution lorsqu'elle est appelée en tête ; en modifiant la valeur de l'entier n on peut interrompre l'exécution du processus à tout moment, et vérifier ainsi que tout se déroule comme prévu.

Afin d'énoncer la spécification de la formule exprimant la transitivité de la relation $\sqsubseteq$, on se donne trois suites d'instructions $(\gamma_i^j)_i$ pour $1 \leqslant j \leqslant 3$, possédant des règles d'exécution identiques, à savoir :

$$\gamma_{i+1}^j \star t \cdot \pi \succ t \star \gamma_i^j \cdot \pi$$

On peut noter que le comportement de l'instruction γ_n^j est celui partagé par tous les termes réalisant une formule de la forme $\neg^{2n}F$ (la preuve est analogue à celle du théorème 3.1.15). Il s'agit donc de la spécification des formules formant les hypothèses du théorème de transitivité. C'est-à-dire que pour expliciter le comportement des termes réalisant une formule de la forme $A_1, \ldots, A_n \to B$, on lui donnera en argument des termes $u_1, \ldots, u_n$ ayant respectivement les comportements typiques de termes réalisant les formules $A_1, \ldots, A_n$; il s'agit intuitivement des « arguments naturels » du terme considéré.

Théorème 3.1.16.
Soit θ réalisant $\forall x \forall y \forall z (x \sqsubseteq y,\ y \sqsubseteq z \to x \sqsubseteq z)$ Alors pour toute pile π, l'exécution de $\theta \star \gamma^1_{n_1} \cdot \gamma^2_{n_2} \cdot \gamma^3_{n_3} \cdot \pi$ se termine sur un processus de la forme :
- *$\gamma^1_0 \star t \cdot \rho$ si $n_1 \leqslant n_2$ et $n_1 \leqslant n_3$,*
- *$\gamma^2_0 \star t \cdot \rho$ si $n_2 < n_1$ et $n_2 \leqslant n_3$,*
- *$\gamma^3_0 \star t \cdot \rho$ si $n_3 < n_1$ et $n_3 < n_2$.*

Interprétation : On peut considérer que chaque instruction γ^j_n code l'entier n. Cette spécification est alors celle des termes qui, étant donné trois entiers, calculent le premier des trois à être minimal parmi eux.

Démonstration. Soit $\perp\!\!\!\perp = < \theta \star \gamma^1_{n_1} \cdot \gamma^2_{n_2} \cdot \gamma^3_{n_3} \cdot \pi >$, pour une pile π quelconque.

1. Si $n_1 \leqslant n_2$ et $n_1 \leqslant n_3$. On considère que θ réalise la formule $n_1{+}1 \sqsubseteq n_1,\ n_1 \sqsubseteq n_1 \to n_1{+}1 \sqsubseteq n_1$. Du lemme 3.1.13 on déduit que $\gamma^2_{n_2}$ réalise $n_1 \sqsubseteq n_1$ et $\gamma^3_{n_3} \Vdash \neg^{2n_1} \top$. Il en résulte $\gamma^1_{n_1} \nVdash (n_1 + 1) \sqsubseteq n_1$, puisque le processus considéré n'est pas dans $\perp\!\!\!\perp$. Le lemme 3.1.14 donne alors $\gamma^1_0 \nVdash \top \to \bot$, c'est-à-dire qu'il existe un terme t et une pile ρ tels que $\gamma^1_0 \star t \cdot \cdot \rho$ apparaisse dans le fil considéré, ce qui termine la démonstration.
2. Si $n_2 < n_1$ et $n_2 \leqslant n_3$. On raisonne de la même manière, en considérant que $\theta \Vdash n_2{+}1 \sqsubseteq n_2{+}2,\ n_2{+}2 \sqsubseteq n_2 \to n_2{+}1 \sqsubseteq n_2$.
3. Si $n_3 < n_1$ et $n_3 < n_1$. On considère dans ce cas que θ réalise $n_3{+}1 \sqsubseteq n_3{+}1,\ n_3{+}1 \sqsubseteq n_3{+}1 \to n_3{+}1 \sqsubseteq n_3{+}1$.

□

Remarque : Lorsqu'on observe l'exécution du processus $\theta \star \gamma^1_{n_1} \cdot \gamma^2_{n_2} \cdot \gamma^3_{n_3} \cdot \pi$, on observe qu'apparaissent en tête *et dans cet ordre* les instructions suivantes : $\gamma^1_{n_1}$, $\gamma^2_{n_2}$, $\gamma^3_{n_3}$, $\gamma^1_{n_1-1}$, $\gamma^2_{n_2-1}$, $\gamma^3_{n_3-1}$... Ceci n'est pas vrai en général : toutes ces instructions apparaissent bien en tête de pile, mais l'ordre peut être différent. Dans le cas où l'on restreint les instructions disponibles à la seule instruction cc, on peut néanmoins démontrer ce résultat, ce qui permet d'interpréter cette spécification comme étant celle de l'ordonnanceur le plus simple qui soit : le *tourniquet*.

Pour $n \geqslant 2$, on peut à l'aide d'une preuve identique à celle du théorème 3.1.11 obtenir une quasi-preuve réalisant la formule

$$\forall x_1 \ldots \forall x_{n+1} (x_1 \sqsubseteq x_2, \ldots, x_n \sqsubseteq x_{n+1} \to x_1 \sqsubseteq x_{n+1}).$$

On obtient alors une spécification analogue à la précédente.

Théorème 3.1.17.
Soit $p \geqslant 2$. La formule $\forall x_1 \ldots \forall x_{p+1} (x_1 \sqsubseteq x_2, \ldots, x_p \sqsubseteq x_{p+1} \to x_1 \sqsubseteq x_{p+1})$ est réalisé par la quasi-preuve

$$(Y)\lambda r \lambda u_1 \lambda u_2 \ldots \lambda u_{p+1} (u_1) \lambda x_1 (u_2) \lambda x_2 (u_3) \ldots (u_p) \lambda x_p (u_{p+1}) (r) x_1 \ldots x_p.$$

Démonstration. Soit θ le terme considéré. Montrons par induction sur l'entier n_1 que le processus $\theta \star \pi$ est dans $\perp\!\!\!\perp$ quels que soient $n_2, \ldots, n_{p+1}$ et π dans $||n_1 \sqsubseteq n_2, \ldots, n_p \sqsubseteq n_{p+1} \to n_1 \sqsubseteq n_{p+1}||$.
Si $n_1 = 0$, il n'existe aucune pile dans cet ensemble, ce qui prouve le résultat.
Sinon, considérons une telle pile π. Celle-ci est alors de la forme $t_1 \cdot \ldots \cdot t_p \cdot u \cdot \rho$ avec $t_i \Vdash n_i \sqsubseteq n_{i+1}$ et $u \Vdash \forall y(n_1 - 1 \sqsubseteq y \to sy \neq n_{p+1})$. Le processus considéré se réduit alors en $t_1 \star \lambda x_1(t_2)\lambda x_2 \ldots (t_p)\lambda x_p(u)(\theta)x_1 \ldots x_p \cdot \rho$. Comme $n_1 \neq 0$, on a $t_1 \Vdash \forall y(n_1 - 1 \sqsubseteq y \to sy \neq n_2) \to \perp$; il nous suffit donc de montrer $\lambda x_1(t_2)\lambda x_2 \ldots (t_p)\lambda x_p(u)(\theta)x_1 \ldots x_p \Vdash \forall y(n_1 - 1 \sqsubseteq y \to sy \neq n_2)$. Si $n_2 = 0$, on à là le résultat. Sinon, on considère un terme t_1' réalisant $n_1 - 1 \sqsubseteq n_2 - 1$, et il nous faut montrer que le processus $\lambda x_1(t_2)\lambda x_2 \ldots (t_p)\lambda x_p(u)(\theta)x_1 \ldots x_p \star t_1' \cdot \rho$ est dans $\perp\!\!\!\perp$. Mais il se réduit en $t_2 \star \lambda x_2(t_3)\lambda x_2 \ldots (t_p)\lambda x_p(u)(\theta)t_1' x_2 \ldots x_p \cdot \rho$, et l'on est ramené à la même discussion : si n_3 est nul il n'y a plus rien à démontrer, et sinon on doit considérer un terme t_2' réalisant $n_2 - 1 \sqsubseteq n_3 - 1$, et montrer que le processus $\lambda x_2(t_3)\lambda x_3 \ldots (t_p)\lambda x_p(u)(\theta)t_1' x_2 \ldots x_p \star t_2' \cdot \rho$ est dans $\perp\!\!\!\perp$. Donc en itérant ce raisonnement, on prouve le résultat si l'un des n_i est nul, et sinon on se ramène à montrer qu'un processus de la forme $u \star (\theta)t_1' t_2' \ldots t_p' \cdot \rho$ est dans $\perp\!\!\!\perp$, avec chacun des t_i' qui réalise $n_i - 1 \sqsubseteq n_{i+1} - 1$. Mais comme $n_{p+1} \neq 0$, on en déduit que u réalise $n_1 - 1 \sqsubseteq n_{p+1} - 1 \to \perp$. Si $n_1 - 1 = 0$ il n'y a plus rien à démontrer, et sinon il suffit de prouver que si u' réalise
$\forall y(n_1 - 2 \sqsubseteq y \to sy \neq n_{p+1} - 1)$, alors le processus $(\theta)t_1' t_2' \ldots t_p' \star u' \cdot \rho$ est dans $\perp\!\!\!\perp$, ce qui découle alors de l'hypothèse d'induction. □

On se donne pour spécifier ce théorème une famille de suites d'instructions $(\gamma_i^j)_i$, pour j entre 1 et $p+1$.

Théorème 3.1.18.
Soit θ réalisant $\forall x_1 \ldots \forall x_{p+1}(x_1 \sqsubseteq x_2, \ldots, x_p \sqsubseteq x_{p+1} \to x_1 \sqsubseteq x_{p+1})$. Alors pour toute pile π, l'exécution de $\theta \star \gamma_{n_1}^1 \cdot \ldots \cdot \gamma_{n_{p+1}}^{p+1} \cdot \pi$ se termine sur un processus de la forme $\gamma_0^j \star t \cdot \rho$, où j est tel que n_j est le premier des entiers $n_1, \ldots, n_{p+1}$ à être minimal parmi eux.

Interprétation : Cette spécification est celle des termes qui permettent de calculer le premier des entiers parmi $p+1 \geqslant 3$ à être le plus petit.

Démonstration. Soit $\pi \in \Pi$ et $\perp\!\!\!\perp = < \theta \star \gamma_{n_1}^1 \cdot \ldots \cdot \gamma_{n_{p+1}}^{p+1} \cdot \pi >$. On appelle n_i le premier parmi les entiers $n_1, \ldots, n_{p+1}$ à être minimal parmi eux ; c'est-à-dire que l'on a $n_i < n_k$ si $k < i$, et $n_i \leqslant n_k$ sinon. On considère alors que θ réalise la formule $F_1, \ldots, F_p \to n_i + 1 \sqsubseteq n_i$, où

$$F_k = \begin{cases} n_i + 1 \sqsubseteq n_i + 1 \text{ si } k < i \\ n_i + 1 \sqsubseteq n_i \text{ si } k = i \\ n_i \sqsubseteq n_i \text{ si } k > i \end{cases}$$

Mais si $k < i$ on a $\gamma^k_{n_k} \Vdash \neg^{2(n_i+1)}\top$, et donc $\gamma^k_{n_k} \Vdash n_i{+}1 \sqsubseteq n_i{+}1$. De même, si $k > i$ on a $\gamma^k_{n_k} \Vdash \neg^{2n_i}\top$, et donc $\gamma^k_{n_k} \Vdash n_i \sqsubseteq n_i$. Enfin, on a $\gamma^{p+1}_{n_{p+1}} \Vdash \neg^{2n_i}\top$, et donc $\gamma^{p+1}_{n_{p+1}} \Vdash \forall y(n_i{+}1 \sqsubseteq y \to sy \neq n_i)$.
Puisque le processus considéré n'est pas dans $\perp\!\!\!\perp$, on en déduit que $\gamma^i_{n_i}$ ne réalise pas $n_i{+}1 \sqsubseteq n_i$. Le lemme 3.1.14 assure alors $\gamma^i_0 \nVdash \top \to \bot$, c'est-à-dire qu'il existe un terme t et une pile ρ tels que $\gamma^i_0 \star t \cdot \rho$ apparaisse dans le fil considéré, ce qui termine la démonstration. □

3.1.3 Totalité

Pour obtenir la spécification analogue à la précédente mais où l'on ne compare que deux entiers, il faut considérer le théorème suivant, qui exprime que la relation $\sqsubseteq$ est totale :

$$\forall x_1 \forall x_2 [(x_1 \sqsubseteq x_2 \to \bot), (x_2 \sqsubseteq x_1 \to \bot) \to \bot].$$

La seule différence dans ce cas, est qu'une quasi-preuve réalisant ce théorème peut indifféremment commencer par mettre en tête son premier ou son second argument. En effet, le mécanisme requis est ici d'extraire le type $\top \to \bot$ de deux suites de longueur paire de négations, qui sont :
- soit toutes les deux de même longueur, pointant sur le type $\top \to \bot$,
- soit telles que l'une est strictement plus courte que l'autre et pointe sur le type $\top \to \bot$.

On peut donc commencer par parcourir l'une ou l'autre indifféremment.

Théorème 3.1.19.
La formule $\forall x_1 \forall x_2 [(x_1 \sqsubseteq x_2 \to \bot), (x_2 \sqsubseteq x_1 \to \bot) \to \bot]$ *est réalisée par chacune des quasi-preuves suivantes :*

1. $(Y)\lambda r \lambda u \lambda v (u) \lambda x (v)(r)x$
2. $(Y)\lambda r \lambda u \lambda v (v) \lambda y (u) \lambda x (r)xy$

On donne dans l'annexe A la présentation en séquents de la démonstration suivante (voir page 157).

Démonstration. Soit θ le premier terme considéré. On prend $\pi \in \Pi$, deux entiers n et p, ainsi que deux termes u et v réalisant respectivement $n \sqsubseteq p \to \bot$ et $p \sqsubseteq n \to \bot$. On va montrer par induction sur n que $\theta \star u \cdot v \cdot \pi \in \perp\!\!\!\perp$. Ce processus se réduit en $u \star \lambda x (v)(\theta)x \cdot \pi$. Il suffit donc de montrer que $\lambda x (v)(\theta)x$ réalise $n \sqsubseteq p$. Si $n = 0$, il n'y a rien à démontrer. Sinon, on considère un terme x réalisant $\forall y(n{-}1 \sqsubseteq y \to sy \neq p)$, et on doit montrer $v \star (\theta)x \cdot \rho \in \perp\!\!\!\perp$ pour toute pile ρ. Il suffit donc de montrer $(\theta)x \Vdash p \sqsubseteq n$. Si $p = 0$ il n'y a rien à

démontrer, et sinon le résultat découle de l'hypothèse d'induction puisqu'alors $x \Vdash n-1 \sqsubseteq p-1 \to \bot$.
Le second point se démontre de manière analogue. □

Théorème 3.1.20.
Soit θ réalisant la formule $\forall x_1 \forall x_2[(x_1 \sqsubseteq x_2 \to \bot), (x_2 \sqsubseteq x_1 \to \bot) \to \bot]$. Alors pour toute pile π, l'exécution de $\theta \star \gamma^1_{n_1} \cdot \gamma^2_{n_2} \cdot \pi$ se termine sur un processus de la forme :
- *$\gamma^1_0 \star t \cdot \rho$ si $n_1 < n_2$,*
- *$\gamma^2_0 \star t \cdot \rho$ si $n_2 < n_1$,*
- *$\gamma^1_0 \star t \cdot \rho$ ou $\gamma^2_0 \star t \cdot \rho$ si $n_1 = n_2$.*

Démonstration. Soit π une pile, posons $\perp\!\!\!\perp =< \theta \star \gamma^1_{n_1} \cdot \gamma^2_{n_2} \cdot \pi >$.
Si $n_1 < n_2$, on a que $\gamma^2_{n_2}$ réalise $\neg^{2(n_1+1)}\top$, soit encore $(n_2 \sqsubseteq n_1 \to \bot)$, et $\theta \Vdash (n_2 \sqsubseteq n_1 \to \bot), (n_1 \sqsubseteq n_2 \to \bot) \to \bot$. Ceci assure que $\gamma^1_{n_1}$ ne réalise pas la formule $(n_1 \sqsubseteq n_2 \to \bot)$ pour ce $\perp\!\!\!\perp$; c'est-à-dire $\gamma^1_{n_1} \nVdash \neg^{2n_1}\top \to \bot$. Le lemme 3.1.14 assure alors $\gamma^1_0 \nVdash \top \to \bot$, c'est-à-dire qu'il existe un terme t et une pile ρ tels que $\gamma^1_0 \star t \cdot \rho$ apparaisse dans le fil considéré, ce qui achève la démonstration dans ce cas.
Si $n_2 < n_1$, on procède de la même manière en constatant que $\gamma^1_{n_1}$ réalise la formule $\neg^{2(n_2+1)}\top = (n_1 \sqsubseteq n_2 \to \bot)$.
Si $n_1 = n_2$, on considère que θ réalise : $(n_1 \sqsubseteq n_1 \to \bot), (n_1 \sqsubseteq n_1 \to \bot) \to \bot$. On a donc l'alternative suivante :

$$\gamma^1_{n_1} \nVdash (n_1 \sqsubseteq n_1 \to \bot) \text{ ou } \gamma^2_{n_2} \nVdash (n_1 \sqsubseteq n_1 \to \bot)$$

Si l'on suppose que la première (respectivement la seconde) assertion est vraie, on obtient grâce au lemme 3.1.14 que l'exécution se termine sur un processus de la forme $\gamma^1_0 \star t \cdot \rho$ (respectivement $\gamma^2_0 \star t \cdot \rho$). □

3.1.4 Bonne fondation

On commence par énoncer le théorème suivant, qui exprime que les modèles génériques sont bien fondés pour le successeur.

Théorème 3.1.21.
Quelle que soit la formule F possédant exactement une variable libre, la quasi-preuve $\overline{Y} = (Y)\lambda r \lambda t \lambda x (\mathsf{cc}) \lambda k (x)(k)(t)(r)tx$ réalise la formule suivante :

$$\forall y(Fy \to Fsy), \forall y(y \simeq 0 \to Fy) \to \forall y Fy$$

Démonstration. Soit t un terme réalisant $\forall y(F(y) \to F(sy))$, et x réalisant $\forall y(y \simeq 0 \to F(y))$, ainsi qu'un entier n et une pile π dans $||F(n)||$. On va prouver par récurrence sur n que le processus $\overline{Y} \star t \cdot x \cdot \pi$ est dans $\perp\!\!\!\perp$. On remarque déjà que ce processus se réduit en $x \star (\mathsf{k}_\pi)(t)(\overline{Y})tx \cdot \pi$.

Si $n = 0$: on a $x \Vdash \top \to F(n)$, ce qui donne le résultat.
Si $n \neq 0$: on a $x \Vdash \bot \to F(n)$, il faut donc vérifier $(\mathsf{k}_\pi)(t)(\overline{Y})tx \Vdash \bot$. Or, pour toute pile ρ, le processus $(\mathsf{k}_\pi)(t)(\overline{Y})tx \star \rho$ se réduit en $t \star (\overline{Y})tx \cdot \pi$, qui est dans $\perp\!\!\!\perp$, puisque $(\overline{Y})tx$ réalise $F(n-1)$ par hypothèse de récurrence. □

Le théorème suivant exprime que pour tout élément d'un modèle générique, il existe un entier du modèle qui possède le même rang que lui.

Théorème 3.1.22.
Soit s un λ-terme pour le successeur.
La formule $\forall x(\forall n(Ent(n), x \sqsubseteq n, n \sqsubseteq x \to \bot) \to \bot)$ est réalisée par la quasi-preuve :

$$\left((\overline{Y})\lambda a\lambda b(a)(\lambda u\lambda n\lambda v\lambda w(u)(s)n\lambda x(x)v\lambda x(x)w)b\right)\lambda u\lambda v(v)\underline{0}uI.$$

Remarque : On réalise aisément la formule
$\forall n\forall p(Ent(n), Ent(p), n \sqsubseteq p, p \sqsubseteq n \to n = p)$ à l'aide d'opérateurs de mise en mémoire et d'un λ-terme comparant deux entiers de Church.

Démonstration. Soit $F(x) = \forall n(Ent(n), x \sqsubseteq n, n \sqsubseteq x \to \bot) \to \bot$. On va réaliser $\forall x F(x)$ en utilisant le théorème précédent.

1. $\lambda u\lambda v(v)\underline{0}uI \Vdash \forall x(x \simeq 0 \to F(x))$. On prend un entier p, u réalisant $p \simeq 0$, v réalisant $\forall n(Ent(n), p \sqsubseteq n, n \sqsubseteq p \to \bot)$ ainsi qu'une pile π. Le processus $\lambda u\lambda v(v)\underline{0}uI \star u \cdot v \cdot \pi$ se réduit en $v \star \underline{0} \cdot u \cdot I \cdot \pi$. Si $p = 0$, on a le résultat considérant que v réalise $Ent(0), 0 \sqsubseteq 0, 0 \sqsubseteq 0 \to \bot$. Sinon, on conclut en considérant $v \Vdash Ent(0), 1 \sqsubseteq, 0 \sqsubseteq 1 \to \bot$.
2. $\lambda a\lambda b(a)(\lambda u\lambda n\lambda v\lambda w(u)(s)n\lambda x(x)v\lambda x(x)w)b \Vdash \forall x(Fx \to Fsx)$.
   On considère un entier p, un terme a réalisant $F(p)$, une pile π ainsi qu'un terme b réalisant $\forall n(Ent(n), sp \sqsubseteq n, n \sqsubseteq sp \to \bot)$. On doit montrer $a \star (\lambda u\lambda n\lambda v\lambda w(u)(s)n\lambda x(x)v\lambda x(x)w)b \cdot \pi \in \perp\!\!\!\perp$. Il suffit donc de montrer $(\lambda u\lambda n\lambda v\lambda w(u)(s)n\lambda x(x)v\lambda x(x)w)b \Vdash \forall n(Ent(n), p \sqsubseteq n, n \sqsubseteq p \to \bot)$. Soient alors n entier, $\bar{n} \Vdash Ent(n)$, $v \Vdash p \sqsubseteq n$, $w \Vdash n \sqsubseteq p$ et une pile π. On doit montrer $\lambda u\lambda n\lambda v\lambda w(u)(s)n\lambda x(x)v\lambda x(x)w \star b \cdot \bar{n} \cdot v \cdot w \cdot \pi \in \perp\!\!\!\perp$. mais ce processus se réduit en $b \star (s)\bar{n} \cdot \lambda x(x)v \cdot \lambda x(x)w \cdot \pi$, et comme b réalise $Ent(sn), sp \sqsubseteq sn, sn \sqsubseteq sp \to \bot$, on peut conclure.

□

Définition 3.1.23.
On note $\sqsubset$ le prédicat défini par : $n \sqsubset p = ||p \sqsubseteq n \to \bot||$.

Lemme 3.1.24. *1. Quels que soient les entiers n et p, $n \sqsubset p$ et $sn \sqsubseteq p$ désignent le même ensemble de piles.*

2. *Si θ est le terme apparaissant au théorème 3.1.11, $\lambda u\lambda v(\mathsf{cc})(\theta)uv$ réalise le théorème*

$$\forall x\forall y\forall z(x \sqsubset y, y \sqsubseteq sz \to x \sqsubseteq z).$$

Démonstration. 1. Trivial.

2. Par hypothèse, si u et v sont des termes réalisant respectivement $m \sqsubset n$ et $n \sqsubseteq sp$, alors $(\theta)uv$ réalise $sm \sqsubseteq sp$ considérant le premier point. Le résultat découle alors du fait que $sm \sqsubseteq sp$ est égal à $||(m \sqsubseteq p \to \bot) \to \bot||$.

□

On peut maintenant réaliser une formule exprimant la bonne fondation de la relation $\sqsubset$ sur les modèles de la réalisabilité.

Théorème 3.1.25.
Soit θ la quasi-preuve donnée au théorème 3.1.11, τ celle donnée au théorème 3.1.22, et P un λ-terme calculant le prédécesseur.
La formule $\forall X \left(\forall x \left(\forall y \{ y \sqsubset x \to \neg Xy \} \to \neg Xx \right) \to \forall x \neg Xx \right)$ *est réalisée par la quasi-preuve*

$$\lambda u \lambda v (\tau) \lambda \bar{n} \lambda x_1 \lambda x_2 (\alpha) u v \bar{n} x_1,$$

où l'on a posé :

$$\alpha = (Y) \lambda r \lambda u \lambda v \lambda \bar{n} \lambda x \left((u) \lambda a \lambda b \left((\bar{n}) \lambda g (r) u b (P) \bar{n} (\mathsf{cc}) (\theta) a x \right) (a) (\theta) x I \right) v.$$

Démonstration. On fixe un prédicat $\Phi : \mathbb{N} \to \mathscr{P}(\Pi)$, un terme u réalisant $\forall x (\forall y \{ y \sqsubset x \to \neg \Phi y \} \to \neg \Phi x)$, un entier m, un terme v réalisant Φm ainsi qu'une pile π. Il nous faut montrer, si Θ désigne le terme considéré, que le processus $\Theta \star u \cdot v \cdot \pi$ est dans $\perp\!\!\!\perp$. Celui-ci se réduisant en $\tau \star \lambda \bar{n} \lambda x_1 \lambda x_2 (\alpha) u v \bar{n} x_1 \cdot \pi$, il suffit de montrer $\lambda \bar{n} \lambda x_1 \lambda x_2 (\alpha) u v \bar{n} x_1 \Vdash \forall n (Ent(n), m \sqsubseteq n, n \sqsubseteq x \to \bot)$.
On va montrer par récurrence sur n que quel que soit cet entier, l'entier m, le terme v réalisant Φm, le terme $\bar{n}$ réalisant $Ent(n)$, le terme t réalisant $m \sqsubseteq n$ ainsi que la pile ρ, le processus $\alpha \star u \cdot v \cdot \bar{n} \cdot t \cdot \rho$ est dans $\perp\!\!\!\perp$, ce qui donnera le résultat. Ce processus se réduisant en

$$u \star \lambda a \lambda b ((\bar{n}) \lambda g (\alpha) u b (P) \bar{n} (\mathsf{cc}) (\theta) a x) (a) (\theta) x I \cdot v \cdot \rho$$

il nous suffit de démontrer que $\lambda a \lambda b ((\bar{n}) \lambda g (\alpha) u b (P) \bar{n} (\mathsf{cc}) (\theta) a t) (a) (\theta) t I$ réalise $\forall y (y \sqsubset m \to \neg \Phi y)$, puisque $v \Vdash \Phi m$. Soit un entier p, un terme a réalisant $p \sqsubset m$ ainsi qu'un terme b réalisant Φp, il nous suffit de démontrer que pour toute pile ρ', $\bar{n} \star \lambda g (\alpha) u b (P) \bar{n} (\mathsf{cc}) (\theta) a t \cdot (a) (\theta) t I \cdot \rho' \in \perp\!\!\!\perp$.
Si $n = 0$, on considère l'élément $\Psi : \mathbb{N} \to \mathscr{P}(\Pi)$ défini par :

$$\Psi(k) = \begin{cases} \Pi \text{ si } k = 0 \\ \top \text{ sinon} \end{cases}$$

Comme $\bar{n} \Vdash \forall x (\Psi x \to \Psi s x), \Psi 0 \to \Psi 0$, il suffit de montrer $(a) (\theta) t I \Vdash \bot$. Considérant que a réalise $m \sqsubseteq p \to \bot$, il suffit de vérifier $(\theta) t I \Vdash m \sqsubseteq p$; mais ceci est évident considérant que $n = 0$ et donc que $I \Vdash n \sqsubseteq p$.
Si $n \neq 0$, on considère l'élément $\Psi : \mathbb{N} \to \mathscr{P}(\Pi)$ défini par :

$$\Psi(k) = \begin{cases} \top \text{ si } k = 0 \\ \Pi \text{ sinon} \end{cases}$$

Il suffit alors de montrer $\lambda g(\alpha)ub(P)\bar{n}(\mathrm{cc})(\theta)at \Vdash \top \to \bot$. Montrons donc $(\alpha)ub(P)\bar{n}(\mathrm{cc})(\theta)at \Vdash \bot$. Le lemme 3.1.24 assure que $(\mathrm{cc})(\theta)at$ réalise $p \sqsubseteq n-1$, ce qui donne le résultat en utilisant l'hypothèse de récurrence. □

Spécification

La spécification de ce théorème peut-être donnée en termes de jeu. On considère pour cela deux suites d'instructions $(\kappa_m)_m$ et $(\gamma_k)_k$ telles que :

$$\gamma_{k+1} \star t \cdot \pi \succ t \star \gamma_k \cdot \pi \text{ pour tout } k \text{ entier}$$

$$\kappa_m \star s^n\underline{0} \cdot \xi_1 \cdot \xi_0 \cdot \pi \succ \begin{cases} \xi_1 \star \gamma_n \cdot \pi \text{ si } n < m \\ \xi_0 \star \gamma_m \cdot \pi \text{ si } n > m \end{cases}$$

Si l'instruction κ_m arrive en tête d'une pile de forme différente, alors l'exécution s'arrête. De même, l'instruction γ_0 ne possède pas de règle d'exécution.

Théorème 3.1.26.
Si θ est un terme réalisant la formule $\forall x \exists n(Ent(n), x \sqsubseteq n, n \sqsubseteq x)$, alors pour tout entier m et toute pile π, l'exécution de $\theta \star (T)\kappa_m \cdot \pi$ s'arrête sur un processus de la forme $\kappa_m \star s^m\underline{0} \cdot \rho$.

Démonstration. Soit $\perp\!\!\!\perp = \{p \in \Lambda_c \times \Pi; p \succ \kappa_m \star s^m\underline{0} \cdot \rho,\ \rho \in \Pi\}$. On va vérifier $(T)\kappa_m \Vdash \forall n(Ent(n), m \sqsubseteq n, n \sqsubseteq m \to \bot)$ pour ce $\perp\!\!\!\perp$, ce qui donnera le résultat. Il suffit de vérifier que pour tout entier n, pour tout terme u réalisant $m \sqsubseteq n$ et pour tout terme v réalisant $n \sqsubseteq m$, le processus $\kappa_m \star s^n\underline{0} \cdot u \cdot v \cdot \rho$ est dans $\perp\!\!\!\perp$. Si $n = m$, c'est trivial. Si $n < m$, ce processus se réduit en $u \star \gamma_n \cdot \pi$, qui est dans $\perp\!\!\!\perp$ considérant $u \Vdash \neg^{2n+1}\top$ et $\gamma_n \Vdash \neg^{2n}\top$ (ce dernier point découle du fait que le $\perp\!\!\!\perp$ choisi est saturé pour toutes les instructions). De même si $n > m$, ce processus se réduit en $v \star \gamma_m \cdot \pi$, qui est dans $\perp\!\!\!\perp$ puisque $v \Vdash \neg^{2m+1}\top$ et $\gamma_m \Vdash \neg^{2m}\top$. □

Interprétation : Tout réalisateur de cette formule implémente donc une stratégie gagnante dans le jeu où le joueur doit deviner un entier par essais successifs, l'opposant lui indiquant après chaque essai si l'entier recherché est supérieur ou inférieur à l'entier proposé. Le terme obtenu au théorème 3.1.22 implémente la stratégie triviale qui consiste à énumérer les entiers dans leur ordre naturel, et donc à ne pas se servir des indications de l'opposant.

3.2 Un prédicat pour l'addition

3.2.1 Définitions

En utilisant le théorème 2.2.40, on peut poser la définition suivante.

Définition 3.2.1.
On appellent $A_{\sqsubseteq}$ et $A_{\sqsupseteq}$ les éléments de $\mathscr{P}(\Pi)^{\mathbb{N}^3}$ respectivement définis par les relations :

$$A_{\sqsubseteq}m,n,p = \Big|\Big|\forall x\left(\forall z\left(A_{\sqsubseteq}x,n,z \to sz \neq p\right) \to sx \neq m\right) \bigwedge \forall y\left(\forall z\left(A_{\sqsubseteq}m,y,z \to sz \neq p\right) \to sy \neq n\right)\Big|\Big|$$

et

$$A_{\sqsupseteq}m,n,p = \Big|\Big|\forall z\left(\forall x\left(A_{\sqsupseteq}x,n,z \to sx \neq m\right) \to sz \neq p\right) \bigvee \forall z\left(\forall y\left(A_{\sqsupseteq}m,y,z \to sy \neq n\right) \to sz \neq p\right)\Big|\Big|.$$

On peut cette fois encore expliciter complètement ces prédicats, ce qui assure leur unicité.

Théorème 3.2.2.
On a les égalités suivantes, quels que soient les entiers l, m et n.

1. – $A_{\sqsubseteq}0,0,p = ||\top \wedge \top||$.
   – $A_{\sqsubseteq}sm,0,0 = ||(\top \to \bot) \wedge \top||$, *le résultat étant symétrique en les deux premiers arguments du prédicat.*
   – $A_{\sqsubseteq}sm,0,sp = ||(\neg^2 A_{\sqsubseteq}m,0,p) \wedge \top||$, *le résultat étant symétrique en les deux premiers arguments du prédicat.*
   – $A_{\sqsubseteq}sm,sn,sp = ||\neg^2 A_{\sqsubseteq}m,sn,p \wedge \neg^2 A_{\sqsubseteq}sm,n,p||$.
   – $A_{\sqsubseteq}sm,sn,0 = ||(\top \to \bot) \wedge (\top \to \bot)||$.
2. – $A_{\sqsupseteq}0,0,sp = ||(\top \to \bot) \vee (\top \to \bot)||$.
   – $A_{\sqsupseteq}sm,0,sp = ||(\neg^2 A_{\sqsupseteq}m,0,p) \vee (\top \to \bot)||$, *le résultat étant symétrique en les deux premiers arguments du prédicat.*
   – $A_{\sqsupseteq}sm,sn,sp = ||\neg^2 A_{\sqsupseteq}m,sn,p \vee \neg^2 A_{\sqsupseteq}sm,n,p||$.
   – $A_{\sqsupseteq}m,n,0 = ||\top \vee \top||$.

Démonstration. Il suffit d'utiliser le lemme 2.2.44. □

Remarque : On peut voir les formules $A_{\sqsubseteq}k,l,k+l$ comme étant des arbres binaires dont les feuilles sont étiquetées par les types $\top$ ou $\top \to \bot$; il s'agit néanmoins d'arbres d'un type particulier puisque toutes les branches maximales y ont la même longueur (à savoir $k+l$).
Les formules $A_{\sqsupseteq}k,l,k+l$ peuvent être vues comme des branches maximales d'un arbre de ce type.

Théorème 3.2.3.
Soient x, y, z des éléments d'un modèle générique $\mathscr{M}$.

1. *$\mathscr{M} \vDash A_{\sqsupseteq} x, y, z$ si et seulement si $rg(x) + rg(y) \geqslant rg(z)$.*
2. *$\mathscr{M} \vDash A_{\sqsubseteq} x, y, z$ si et seulement si $rg(x) + rg(y) \leqslant rg(z)$.*

Démonstration. 1. On prouve le résultat par récurrence sur le rang de z.
Si $rg(z) = 0$: On doit montrer que pour tout x, y dans $\mathscr{M}$, l'une des deux formules suivantes est satisfaite :

$$\forall z'(\forall x'(A_{\sqsupseteq} x', y, z' \to sx' \neq x) \to sz' \neq z)$$

$$\text{ou } \forall z'(\forall y'(A_{\sqsupseteq} x, y', z' \to sy' \neq y) \to sz' \neq z).$$

Mais pour tout élément z' du modèle, la formule $sz' \neq z$ est satisfaite, c'est donc que les deux formules ci-dessus le sont également.
Si $rg(z) > 0$: Supposons dans un premier temps avoir deux éléments x et y du modèle tels que : $rg(x) + rg(y) < rg(z)$, et tels que $\mathscr{M} \vDash A_{\sqsupseteq} x, y, z$. C'est donc que l'une au moins des deux formules suivantes est satisfaite :

$$\forall z'(\forall x'(A_{\sqsupseteq} x', y, z' \to sx' \neq x) \to sz' \neq z)$$

$$\text{ou } \forall z'(\forall y'(A_{\sqsupseteq} x, y', z' \to sy' \neq y) \to sz' \neq z).$$

Supposons que ce soit la première (on raisonnerait de même sinon). Soit z' l'élément de $\mathscr{M}$ tel que $\mathscr{M} \vDash sz' = z$; c'est donc que la formule $\forall x'(A_{\sqsupseteq} x', y, z' \to sx' \neq x)$ n'est pas satisfaite, ce qui implique $rg(x) > 0$. Soit donc x' l'élément de $\mathscr{M}$ tel que $\mathscr{M} \vDash sx' = x$. On a alors
$\mathscr{M} \vDash A_{\sqsupseteq} x', y, z'$, ce qui contredit l'hypothèse de récurrence.
Réciproquement, supposons avoir deux éléments x et y de $\mathscr{M}$ tels que $rg(x) + rg(y) \geqslant rg(z)$. Supposons par exemple $rg(x) > 0$. On va montrer $\mathscr{M} \vDash \forall z'(\forall x'(A_{\sqsupseteq} x', y, z' \to sx' \neq x) \to sz' \neq z)$, ce qui donnera le résultat. Soit z' l'unique élément de $\mathscr{M}$ tel que $\mathscr{M} \vDash sz' = z$; on doit montrer $\mathscr{M} \nvDash \forall x'(A_{\sqsupseteq} x', y, z' \to sx' \neq x)$. Mais si x' est l'élément de $\mathscr{M}$ tel que $\mathscr{M} \vDash sx' = x$, on a par hypothèse de récurrence $\mathscr{M} \vDash A_{\sqsupseteq} x', y, z'$, ce qui permet de conclure.

2. Le second résultat se prouve par récurrence sur le rang de x.
Si $rg(x) = 0$: On va d'abord montrer $\mathscr{M} \vDash A_{\sqsubseteq} x, y, z$ quels que soient les éléments y et z tels que $rg(y) \leqslant rg(z)$. On procède pour cela par récurrence sur le rang de y. Si $rg(y) = 0$, on doit montrer $\mathscr{M}$ satisfait les formules

$$\forall x'(\forall z'(A_{\sqsubseteq} x', y, z' \to sz' \neq z) \to sx' \neq x)$$

$$\text{et } \forall y'(\forall z'(A_{\sqsubseteq} x, y', z' \to sz' \neq z) \to sy' \neq y).$$

Pour tout w de $\mathscr{M}$, on a $\mathscr{M} \vDash sw \neq x$ et $\mathscr{M} \vDash sw \neq y$. Donc $\mathscr{M}$ satisfait fatalement les deux formules ci-dessus. Si $rg(y) > 0$, on a pour ces mêmes raisons que $\mathscr{M} \vDash \forall x'(\forall z'(A_{\sqsubseteq} x', y, z' \to sz' \neq z) \to sx' \neq x)$ pour tout

élément z de $\mathscr{M}$. On doit encore montrer que si y' est tel que $\mathscr{M} \vDash sy' = y$, alors $\mathscr{M} \nvDash \forall z'(A_{\sqsubseteq}x, y', z' \to sz' \neq z)$. Il suffit de montrer $\mathscr{M} \vDash A_{\sqsubseteq}x, y', z'$ si z' est tel que $\mathscr{M} \vDash sz' = z$, ce qui découle de l'hypothèse de récurrence. Réciproquement, on doit vérifier que si $rg(z) \geqslant rg(y)$, alors $\mathscr{M}$ satisfait la formule $\forall y'(\forall z'(A_{\sqsubseteq}x, y', z' \to sz' \neq z) \to sy' \neq y)$. Il suffit de montrer que si y' est tel que $\mathscr{M} \vDash sy' = y$, alors $\mathscr{M} \nvDash \forall z'(A_{\sqsubseteq}x, y', z' \to sz' \neq z)$. Soit z' l'élément de $\mathscr{M}$ tel que $sz' = z$, il vient par hypothèse de récurrence $\mathscr{M} \vDash A_{\sqsubseteq}x, y', z'$, ce qui donne le résultat.

Si $rg(x) > 0$: Supposons dans un premier temps avoir deux éléments x et y du modèle tels que $rg(x) + rg(y) > rg(z)$, et tels que $\mathscr{M} \vDash A_{\sqsubseteq}x, y, z$. C'est donc que la formule $\forall x'(\forall z'(A_{\sqsubseteq}x', y, z' \to sz' \neq z) \to sx' \neq x)$ est satisfaite. Si x' est l'élément de $\mathscr{M}$ tel que $sx' = x$, il vient que $\mathscr{M}$ ne satisfait pas $\forall z'(A_{\sqsubseteq}x', y, z' \to sz' \neq z)$. C'est donc que $rg(z) > 0$. Soit alors z' l'élément de $\mathscr{M}$ tel que $sz' = z$. On a alors $\mathscr{M} \vDash A_{\sqsubseteq}x', y, z'$, ce qui contredit l'hypothèse de récurrence.

Réciproquement, soient x et y tels que $rg(x) + rg(y) \leqslant rg(z)$. On doit montrer que $\mathscr{M}$ satisfait les formules

$$\forall x'(\forall z'(A_{\sqsubseteq}x', y, z' \to sz' \neq z) \to sx' \neq x)$$

$$\text{et } \forall y'(\forall z'(A_{\sqsubseteq}x, y', z' \to sz' \neq z) \to sy' \neq y).$$

La première formule est évidemment satisfaite si $rg(x) = 0$, et sinon on considère l'élément x' tel que $\mathscr{M} \vDash sx' = x$. Il suffit de montrer que $\mathscr{M}$ ne satisfait pas $(\forall z'(A_{\sqsubseteq}x', y, z' \to sz' \neq z)$. Mais si z' est l'élément tel que $\mathscr{M} \vDash sz' = z$, on a par hypothèse de récurrence $\mathscr{M} \vDash A_{\sqsubseteq}x, y', z'$, ce qui donne le résultat. On raisonnerait de même pour la seconde formule.

□

Spécification des formules atomiques écrites avec $A_{\sqsubseteq}$ ou $A_{\sqsupseteq}$

Commençons par donner les deux suites d'instructions qui nous serviront pour les spécifications à venir. On considère trois suites d'instructions $(\alpha_{i,j})$, $(\alpha^1_{i,j})$ et $(\alpha^0_{i,j})$, définies par les règles d'exécution suivantes :

- $\alpha_{i,j} \star t \cdot \pi \succ t \star \alpha^1_{i,j} \cdot \alpha^0_{i,j} \cdot \pi$.
- $\alpha^1_{i+1,j} \star t \cdot \pi \succ t \star \alpha_{i,j} \cdot \pi$.
- $\alpha^0_{i,j+1} \star t \cdot \pi \succ t \star \alpha_{i,j} \cdot \pi$.

On suppose de plus que si l'une des ces instructions arrive en tête d'une pile vide, l'exécution s'arrête.

Les autres instructions de cette suite sont de plus supposées ne pas avoir de règle d'exécution.

On considère par ailleurs trois suites d'instructions $(\beta_{i,j})$, $(\beta^1_{i,j})$ et $(\beta^0_{i,j})$ définies par

- $\beta_{i+1,j} \star t_1 \cdot t_0 \cdot \pi \succ t_1 \star \beta^1_{i+1,j} \cdot \pi$,
- $\beta^1_{i+1,j} \star t \cdot \pi \succ t \star \beta_{i,j} \cdot \pi$,
- $\beta_{0,j+1} \star t_1 \cdot t_0 \cdot \pi \succ t_0 \star \beta^0_{0,j+1} \cdot \pi$,
- $\beta^0_{0,j+1} \star t \cdot \pi \succ t \star \beta_{0,j} \cdot \pi$.

On suppose de plus que $\beta_{0,0} \star t_1 \cdot t_0 \cdot \pi$ se réduit en $t_1 \star \beta^1_{0,0} \cdot \rho$ ou en $t_0 \star \beta^0_{0,0} \cdot \rho$. On suppose encore une fois que les autres instructions de cette suite n'ont pas de règle d'exécution.

Remarque : Les instructions $\alpha_{k,l}$ et $\beta_{k,l}$ codent chacune le couple d'entiers (k,l), mais d'une manière différente.

Théorème 3.2.4.
Soit θ un terme réalisant la formule $A_{\sqsupseteq}(m,n,m+n)$ pour tout $\perp\!\!\!\perp$. Pour toute pile π, l'exécution de $\theta \star \alpha^1_{k,l} \cdot \alpha^0_{k,l} \cdot \pi$ se termine sur un processus de la forme :

- *$\alpha^\tau_{0,0} \star t \cdot \rho$ si $k = 0$ et $l = 0$, avec τ égal à 0 ou à 1.*
- *$\alpha^\tau_{\tilde{k},\tilde{l}} \star t \cdot \rho$ si $k < m$ ou $l < n$, avec τ égal à 0 ou à 1, $0 \leqslant \tilde{k} \leqslant k$ et $0 \leqslant \tilde{l} \leqslant l$, l'un au moins de ces deux entiers étant égal à 0.*

Un tel résultat est inutile du point de vue de la programmation., puisque l'on ne dispose d'aucune information sur ce qui se passe lorsque $k \geqslant m$ et $l \geqslant n$.

Démonstration. Soit $p = m + n$; supposons être dans l'un des trois cas suivants : $k = l = 0$, $k < m$ ou $l < n$.
Si $\perp\!\!\!\perp =< \theta \star \alpha^1_{k,l} \cdot \alpha^0_{k,l} \cdot \pi >$ on a, considérant la formule réalisée par θ, que :

$$\alpha^1_{k,l} \not\Vdash \forall z \left\{ \forall x (A_{\sqsupseteq} x, n, z \to sx \neq m) \to sz \neq p \right\} \to \{\pi\}$$

ou

$$\alpha^0_{k,l} \not\Vdash \forall z \left\{ \forall y (A_{\sqsupseteq} m, y, z \to sy \neq n) \to sz \neq p \right\} \to \{\pi\}.$$

On va montrer par induction sur $k + l$ que cette alternative donne le résultat.
Supposons par exemple que la première assertion soit vraie (on raisonnerait de même sinon). Il existe donc un terme t réalisant
$\forall z[\forall x(A_{\sqsupseteq} x, n, z \to sx \neq m) \to sz \neq p]$ tels que $\alpha^1_{k,l} \star t \cdot \pi$ apparaisse dans le fil considéré. Ceci prouve le résultat dans chacun des trois cas suivants : $k = l = 0$, $k = 0 < m$ et ($k = 0 = m$ et $l > n$).
Si $k \neq 0$, ce processus se réduit en $t \star \alpha_{k-1,l} \cdot \pi$. On a alors que $\alpha_{k-1,l}$ ne réalise pas la formule $\forall x(A_{\sqsupseteq} x, n, p-1 \to sx \neq m)$, ce qui assure $m \neq 0$ (car sinon la valeur de vérité de cette formule est égale à $\top$). Il en suit l'existence d'un terme θ_1 réalisant $A_{\sqsupseteq}(m-1, n, p-1)$ et d'une pile ρ tels que $\alpha_{k-1,l} \star \theta_1 \cdot \rho$ apparaisse dans le fil considéré. Or ce processus se réduit en $\theta_1 \star \alpha^1_{k-1,l} \cdot \alpha^0_{k-1,l} \cdot \rho$. Il en résulte :

$\alpha^1_{k-1,l} \not\Vdash \forall z \Big\{ \forall x(A_{\sqsupseteq} x, n, z \to sx \neq m-1) \to sz) \neq p-1 \Big\} \to \{\rho\}$
ou
$\alpha^0_{k-1,l} \not\Vdash \forall z \Big\{ \forall y(A_{\sqsupseteq} m, y, z \to sy \neq n-1) \to sz \neq p-1 \Big\} \to \{\rho\}$, ce qui donne le résultat par hypothèse d'induction. □

Théorème 3.2.5.
Soit θ un terme réalisant la formule $A_{\sqsubseteq} k', l', k'+l'$ pour tout $\perp\!\!\!\perp$. Si $k < k'$ et $l < l'$, alors pour toute pile π, l'exécution de $\theta \star \beta_{k,l} \cdot \pi$ se termine sur un processus de la forme $\beta^\tau_{0,0} \star t \cdot \rho$ avec τ égal à 0 ou à 1.

Démonstration. Supposons $k' \leqslant k$, $l' \leqslant l$ et posons $p = k' + l'$.
Si $\perp\!\!\!\perp = < \theta \star \beta_{k,l} \cdot \pi >$, on obtient que $\beta_{k,l}$ ne réalise pas la formule suivante
$\forall x(\forall z(A_{\sqsubseteq} x, l', z \to sz \neq p) \to sx \neq k')$,
$\forall y(\forall z(A_{\sqsubseteq} k', y, z \to sz \neq p) \to sy \neq l') \to \{\pi\}$.
On va démontrer par induction sur $k+l$, que cette dernière assertion donne le résultat.
Il existe donc deux termes t_1, t_0 réalisant respectivement
$\forall y(\forall z(A_{\sqsubseteq} x, l', z \to sz \neq p) \to sx \neq k')$
et $\forall y(\forall z(A_{\sqsubseteq} k', y, z \to sz \neq p) \to sy \neq l')$ tels que $\beta_{k,l} \star t_1 \cdot t_0 \cdot \pi$ soit dans le fil considéré.
Si $k = l = 0$, supposons par exemple que ce processus se réduit en $t_1 \star \beta^1_{0,0} \cdot \pi$. On en déduit $\beta^1_{0,0} \not\Vdash \forall z(A_{\sqsubseteq} k'-1, l', z \to sz \neq p)$. Il existe donc un terme t réalisant $A_{\sqsubseteq}(k'-1, l', p-1)$ et une pile π' tel que $\beta^1_{0,0} \star t \cdot \pi'$ soit dans le fil considéré, ce qui donne le résultat dans ce cas.
Si $k = 0$ et $l \neq 0$, ce processus se réduit en $t_0 \star \beta^0_{0,l} \cdot \pi$. De $l' > l > 0$, on déduit que $\beta^0_{0,l}$ ne réalise pas $\forall z(A_{\sqsubseteq} k', l'-1, z \to sz \neq p)$. C'est donc que $p \neq 0$, et qu'il existe un terme θ_1 réalisant $A_{\sqsubseteq} k', l'-1, p-1$ et une pile ρ tels que $\beta^0_{0,l} \star \theta_1 \cdot \rho$ appartienne au fil considéré. Ce processus se réduit en $\theta_1 \star \beta_{0,l-1} \cdot \rho$, qui appartient donc lui aussi au fil. C'est donc que $\beta_{0,l-1}$ ne réalise pas la formule
$\forall x(\forall z(A_{\sqsubseteq} x, l'-1, z \to sz \neq p-1) \to sx \neq k')$,
$\forall y(\forall z(A_{\sqsubseteq} m, y, z \to sz \neq p-1) \to sy \neq n-1) \to \{\rho\}$,
ce qui donne le résultat par hypothèse d'induction.
Si $k \neq 0$, ce processus se réduit en $t_1 \star \beta^1_{k,l} \cdot \rho'$. De $k' > k > 0$, on déduit que $\beta^1_{k,l}$ ne réalise pas $\forall z(A_{\sqsubseteq} k'-1, l', z \to sz \neq p)$. C'est donc que $p \neq 0$, et qu'il existe un terme θ_1 réalisant $A_{\sqsubseteq} k', l'-1, p-1$ et une pile ρ tels que $\beta^1_{k,l} \star \theta_1 \cdot \rho$ appartienne au fil considéré. Ce processus se réduit en $\theta_1 \star \beta_{k-1,l} \cdot \rho'$, qui appartient donc lui aussi au fil. C'est donc que $\beta_{k-1,l}$ ne réalise pas la formule
$\forall x(\forall z(A_{\sqsubseteq} x, l', z) \to sz \neq p-1) \to sx \neq k'-1)$,
$\forall y(\forall z(A_{\sqsubseteq} k'-1, y, z \to sz \neq p-1) \to sy \neq l') \to \{\rho\}$
ce qui donne le résultat par hypothèse d'induction. □

On donne encore les deux résultats suivants qui seront utilisés dans la suite.

Lemme 3.2.6.
Soit $p \in \Lambda_c \star \Pi$ et $\perp\!\!\!\perp =< p >$. Alors $\beta_{k,l} \Vdash A_{\sqsupseteq} k, l', p$ si $p \leqslant k + l'$ et $l' \leqslant l$.
De même, $\beta_{k,l} \Vdash A_{\sqsupseteq} k', 0, p$ si $p \leqslant k' \leqslant k$.

Remarque : L'instruction $\beta_{k',l'}$ est un réalisateur particulier de cette formule, puisqu'elle va d'abord décrémenter l'entier k' jusqu'à 0, avant de décrémenter l'entier l' à son tour. C'est cette définition qui permet de donner la spécification la plus simple possible des formules comportant le prédicat $A_{\sqsupseteq}$.

Démonstration. Démontrons le premier point par induction sur p, l'autre preuve étant analogue. On considère donc un ensemble Φ de piles, $\pi \in \Phi$ et deux termes t_1, t_0 réalisant respectivement
$\forall z \left\{ \forall x (A_{\sqsupseteq} x, l', z \to sx \neq k) \to sz \neq p \right\} \to \Phi$
et $\forall z \left\{ \forall y (A_{\sqsupseteq} k, y, z \to sy \neq l') \to sz \neq p \right\} \to \Phi$. Montrons $\beta_{k,l} \star t_1 \cdot t_0 \cdot \pi \in \perp\!\!\!\perp$.
Si $p = 0$, on a que t_1 et t_0 réalisent $\top \to \Phi$, ce qui donne le résultat considérant la règle d'exécution de $\beta_{k,l}$ et le fait que $\perp\!\!\!\perp$ est saturé pour celle-ci (Cf le théorème 2.2.2).
Sinon, supposons par exemple $k \neq 0$ (on raisonnerait de même sinon) ; ce processus se réduit alors en $t_1 \star \beta^1_{k,l} \cdot \pi$, et il suffit de montrer que $\beta^1_{k,l}$ réalise la formule $\forall x (A_{\sqsupseteq} x, l', p - 1 \to sx \neq k) \to \perp$. On prend alors un terme t réalisant $\forall x (A_{\sqsupseteq} x, l', p - 1 \to sx \neq k)$, une pile ρ et l'on doit montrer que
$\beta^1_{k,l} \star t \cdot \rho \in \perp\!\!\!\perp$. Mais ce processus se réduit en $t \star \beta_{k-1,l} \cdot \rho$, ce qui donne le résultat par induction. □

Lemme 3.2.7.
Soit $p \in \Lambda_c \star \Pi$, et $\perp\!\!\!\perp =< p >$. Alors $\alpha_{k,l} \Vdash A_{\sqsubseteq} k', l', p$ si $k' \leqslant k$, $l' \leqslant l$ et $k' + l' \leqslant p$.

Démonstration. On va démontrer le résultat par induction sur p.
On considère un ensemble Φ de piles, $\pi \in \Phi$ et un terme t réalisant
$[\forall x (\forall z (A_{\sqsubseteq} x, l', z \to sz \neq p) \to sx \neq k')]$,
$[\forall y (\forall z (A_{\sqsubseteq} k', y, z) \to sz \neq p) \to sy \neq l')] \to \Phi$. Il faut montrer $\alpha_{k,l} \star t \cdot \pi \in \perp\!\!\!\perp$.
Ce processus se réduisant en $t \star \alpha^1_{k,l} \cdot \alpha^0_{k,l} \cdot \pi$, il suffit donc de montrer
$\alpha^1_{k,l} \Vdash \forall x (\forall z (A_{\sqsubseteq} x, l', z) \to sz \neq p) \to sx \neq k')$ et que
$\alpha^0_{k,l} \Vdash \forall y (\forall z (A_{\sqsubseteq} k', y, z) \to sz \neq p) \to sy \neq k')$.
Montrons la première assertion, la preuve de la seconde étant similaire. Il n'y a rien à montrer si $k' = 0$, et c'est notamment le cas si $p = 0$. Sinon, on considère un terme u réalisant $A_{\sqsubseteq} k' - 1, l', p - 1 \to \perp$. On doit montrer que pour toute pile ρ, $\alpha^1_{k,l} \star u \cdot \rho \in \perp\!\!\!\perp$. Or ce processus se réduit en $u \star \alpha_{k-1,l} \cdot \rho$, ce qui donne le résultat par hypothèse d'induction. □

3.2.2 Fonctionnalité

Définition 3.2.8.
On note $\dot{1}$ *le terme* $\lambda x \lambda y\ x$, *et* $\dot{0}$ *le terme* $\lambda x \lambda y\ y$.

Lemme 3.2.9.
Les séquents suivants sont dérivables :
i) $\vdash \dot{1} : \forall X \forall Y (X, Y \to X)$,
ii) $\vdash \dot{0} : \forall X \forall Y (X, Y \to Y)$.

Démonstration. On considère les deux dérivations suivantes :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{x : X, y : Y \vdash x : X}{x : X \vdash \lambda y\ x : Y \to X}}{\vdash \lambda x \lambda y\ x : X, Y \to X}}{\vdash \lambda x \lambda y\ x : \forall Y (X, Y \to X)}}{\vdash \lambda x \lambda y\ x : \forall X \forall Y (X, Y \to X)}$$

et

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{x : X, y : Y \vdash y : Y}{x : X \vdash \lambda y\ y : Y \to Y}}{\vdash \lambda x \lambda y\ x : X, Y \to Y}}{\vdash \lambda x \lambda y\ y : \forall Y (X, Y \to Y)}}{\vdash \lambda x \lambda y\ y : \forall X \forall Y (X, Y \to Y)}$$

□

Théorème 3.2.10.
La formule $\forall x \forall x' \forall x'' \forall z (A_{\sqsubseteq}(x', x'', x),\ A_{\sqsupseteq}(x', x'', z) \to z \sqsubseteq x)$ *est réalisée par la quasi-preuve suivante :*

$$(Y)\lambda r \lambda a \lambda b \lambda c \left((b)\lambda x(x)\lambda y((a)\dot{1})\lambda z(c)(r)zy\right) \lambda x(x)\lambda y((a)\dot{0})\lambda z(c)(r)zy.$$

Considérons un modèle générique $\mathscr{M}$, et R la relation binaire sur $\mathscr{M}$ qui exprime le fait que deux éléments ont le même rang. La réalisation de cette formule permet, en utilisant deux fois la quasi-preuve associée, de réaliser sans difficulté une formule exprimant que le prédicat A obtenu par la conjonction de $A_{\sqsubseteq}$ et de $A_{\sqsupseteq}$ induit sur le quotient $\mathscr{M}/R$ un prédicat fonctionnel en son troisième argument. On définit pour cela une relation $x \equiv y$ par la formule $x \sqsubseteq y \wedge y \sqsubseteq x$. Le prédicat $\equiv$ sera alors interprété dans chaque modèle générique $\mathscr{M}$ par la relation binaire R considérée. Et en employant deux fois le terme ci-dessus, on réalise la formule :

$$\forall x \forall x' \forall x'' \forall z (A(x', x'', x), A(x', x'', z) \to z \equiv x).$$

C'est-à-dire que le comportement d'un terme réalisant la fonctionnalité du prédicat A est essentiellement celui d'un couple de termes réalisant la formule du

théorème précédent.

La preuve suivante est présentée en séquents dans l'annexe A (voir page 161).

Démonstration du théorème 3.2.10. Soit θ la quasi-preuve considérée. Nous allons montrer par induction sur q que quels que soient les entiers m, n et p, les termes a et b réalisant respectivement les formules $A_{\sqsubseteq}(m,n,p)$ et $A_{\sqsupseteq}(m,n,q)$, ainsi que la pile $\pi \in ||q \sqsubseteq p||$, le processus $\theta \star a \cdot b \cdot \pi$ est dans $\perp\!\!\!\perp$.
Si $q = 0$, on a $||q \sqsubseteq p|| = \top$, et le résultat est alors trivial. Sinon, π est de la forme $c \cdot \rho$ avec c qui réalise $\forall y(q-1 \sqsubseteq y \to sy \neq p-1)$. Le processus considéré se réduit alors en

$$b \star \lambda x(x)\lambda y((a)\dot{1})\lambda z(c)(\theta)zy \cdot \lambda x(x)\lambda y((a)\dot{0})\lambda z(c)(\theta)zy \cdot \rho.$$

Il suffit donc de démontrer :
i) $\lambda x(x)\lambda y((a)\dot{1})\lambda z(c)(\theta)zy \Vdash \forall z \left(\forall x(A_{\sqsupseteq}x,n,z \to sx \neq m) \to sz \neq q\right) \to \{\rho\}$
ii) $\lambda x(x)\lambda y((a)\dot{0})\lambda z(c)(\theta)zy \Vdash \forall z \left(\forall y(A_{\sqsupseteq}m,y,z \to sy \neq n) \to sz \neq q\right) \to \{\rho\}$
Nous allons montrer le point *i)*, la preuve de la seconde assertion étant analogue. On prend x réalisant $\forall z(\forall x(A_{\sqsupseteq}x,n,z \to sx \neq m) \to sz \neq q)$, et on doit montrer que le processus $\lambda x(x)\lambda y((a)\dot{1})\lambda z(c)(\theta)zy \star x \cdot \rho$ est dans $\perp\!\!\!\perp$. Celui-ci se réduit en $x \star \lambda y((a)\dot{1})\lambda z(c)(\theta)zy \star x \cdot \rho$, et comme x réalise la formule $\forall x(A_{\sqsupseteq}(x,n,q-1) \to sx \neq m) \to \perp$, il suffit de montrer que $\lambda y((a)\dot{1})\lambda z(c)(\theta)zy$ réalise $\forall x(A_{\sqsupseteq}x,n,q-1 \to sx \neq m)$. Si $m = 0$ il n'y a rien à démontrer ; sinon on considère un terme y réalisant $A_{\sqsupseteq}(m-1,n,q-1)$, et l'on doit montrer $\lambda y((a)\dot{1})\lambda z(c)(\theta)zy \star y \cdot \rho' \in \perp\!\!\!\perp$ pour toute pile ρ'. Or ce processus se réduit en $(a)\dot{1} \star \lambda z(c)(\theta)zy\rho'$. Mais considérant que a réalise une formule de la forme $\forall X \left(\left(A_1, A_2 \to X \right) \to X \right)$, et que $\dot{1} \Vdash A_1, A_2 \to A_1$, il vient $(a)\dot{1}$ réalise $\forall x(\forall z(A_{\sqsubseteq}x,n,z \to sz \neq p) \to sx \neq m)$. Puisque $m \neq 0$, il suffit de montrer que $\lambda z(c)(\theta)zy$ réalise $\forall z(A_{\sqsubseteq}m-1,n,z \to sz \neq p)$. Si $p = 0$ il n'y a rien à démontrer. Sinon, il suffit de prouver que si z réalise $A_{\sqsubseteq}(m-1,n,p-1)$, le processus $c \star (\theta)zy \cdot \pi'$ est dans $\perp\!\!\!\perp$ pour toute pile π'. Mais comme c réalise $q-1 \sqsubseteq p-1 \to \perp$, l'hypothèse d'induction donne le résultat.

□

Observons le comportement de cette quasi-preuve. On note θ ce terme, et l'on considère les instructions $\alpha_{k,l}$, $\beta_{k',l'}$ et γ_n définies précédemment. On suppose que les entiers n, k et k' sont non nuls.

$$\begin{aligned}
\theta \star \alpha_{k,l} \cdot \beta_{k',l'} \cdot \gamma_n \cdot \pi &\succ \beta_{k',l'} \star \lambda x(x)\lambda y((\alpha_{k,l})\dot{1})\lambda z(\gamma_n)(t)zy \cdot \lambda x(x)\lambda y((\alpha_{k,l})\dot{0})\lambda z(\gamma_n)(t)zy \cdot \pi \\
&\succ \lambda x(x)\lambda y((\alpha_{k,l})\dot{1})\lambda z(\gamma_n)(t)zy \star \beta^1_{k',l'} \cdot \pi \\
&\succ \beta^1_{k',l'} \star \lambda y((\alpha_{k,l})\dot{1})\lambda z(\gamma_n)(t)zy \cdot \pi \\
&\succ (\alpha_{k,l})\dot{1} \star \lambda z(\gamma_n)(t)z\beta_{k'-1,l'} \cdot \pi \\
&\succ \lambda x\lambda y\ x \star \alpha^1_{k,l} \cdot \alpha^0_{k,l} \cdot \lambda z(\gamma_n)(t)z\beta_{k'-1,l'} \cdot \pi \\
&\succ \alpha^1_{k,l} \star \lambda z(\gamma_n)(t)z\beta_{k'-1,l'} \cdot \pi \\
&\succ \gamma_n \star (\theta)\alpha_{k-1,l}\beta_{k'-1,l'} \cdot \pi \\
&\succ \theta \star \alpha_{k-1,l} \cdot \beta_{k'-1,l'} \cdot \gamma_{n-1} \cdot \pi
\end{aligned}$$

Si l'on suppose l, l' et n non nuls, on obtient l'exécution suivante :

$$\begin{aligned}
\theta \star \alpha_{0,l} \cdot \beta_{0,l'} \cdot \gamma_n \cdot \pi &\succ \beta_{0,l'} \star \lambda x(x)\lambda y((\alpha_{0,l})\dot{1})\lambda z(\gamma_n)(t)zy \cdot \lambda x(x)\lambda y((\alpha_{0,l})\dot{0})\lambda z(\gamma_n)(t)zy \cdot \pi \\
&\succ \lambda x(x)\lambda y((\alpha_{0,l})\dot{0})\lambda z(\gamma_n)(t)zy \star \beta^0_{0,l'} \cdot \pi \\
&\succ \beta^0_{0,l'} \star \lambda y((\alpha_{0,l})\dot{0})\lambda z(\gamma_n)(t)zy \cdot \pi \\
&\succ (\alpha_{0,l})\dot{0} \star \lambda z(\gamma_n)(t)z\beta_{0,l'-1} \cdot \pi \\
&\succ \lambda x\lambda y\ y \star \alpha^1_{k,l} \cdot \alpha^0_{k,l} \cdot \lambda z(\gamma_n)(t)z\beta_{0,l'} \cdot \pi \\
&\succ \alpha^0_{0,l} \star \lambda z(\gamma_n)(t)z\beta_{0,l'-1} \cdot \pi \\
&\succ \gamma_n \star (\theta)\alpha_{0,l-1}\beta_{0,l'-1} \cdot \pi \\
&\succ \theta \star \alpha_{0,l-1} \cdot \beta_{0,l'-1} \cdot \gamma_{n-1} \cdot \pi
\end{aligned}$$

Néanmoins, on obtiendra une spécification plus pauvre que ce que cette exécution laisse espérer. Le terme obtenu ci-dessus peut être utilisé pour effectuer des recherches dans des arbres binaires. En effet, en modifiant le comportement des suites α et β, on peut obtenir des instructions codant respectivement un arbre binaire et une branche d'un tel arbre. Au cours de l'exécution du processus $\theta \star \alpha_{k,l} \cdot \beta_{k',l'} \cdot \gamma_n \cdot \pi$, chacun des choix fait par l'instruction courante issue de la suite β détermine laquelle des deux instructions courantes issues de la suite α est conservée pour la suite de l'exécution. Néanmoins, tous les réalisateurs de ce théorème n'ont pas un comportement aussi régulier.

Théorème 3.2.11.
Soit θ réalisant la formule $\forall x \forall x' \forall x'' \forall z (A_{\sqsubseteq}(x', x'', x),\ A_{\sqsupseteq}(x', x'', z) \rightarrow z \sqsubseteq x)$ Alors, pour toute pile π, l'exécution de $\theta \star \alpha_{k,l} \cdot \beta_{k',l'} \cdot \gamma_n \cdot \pi$ se termine sur un processus de la forme :

- *$\beta^{\tau}_{0,0} \star t \cdot \rho$ avec τ égal à 0 ou 1, si $k' \leqslant k$, $l' \leqslant l$ et $k' + l' \leqslant n$.*
- *$\alpha^{1}_{0,l} \star t \cdot \rho$ si $k' > k$ et $k \leqslant n$.*
- *$\alpha^{0}_{\bar{k},0} \star t \cdot \rho$ si $k' \leqslant k$, $l' > l$ et $k' + l \leqslant n$, où $\bar{k}$ est compris entre $k - k'$ et k.*
- *$\gamma_0 \star t \cdot \rho$ sinon.*

Interprétation : Cette spécification est celle des termes qui, étant donné deux couples d'entiers (k', l') et (k, l), permettent de calculer si les composantes du premier couple sont toutes les deux inférieures à celle du second, ou sinon de donner la première composante du premier couple à être strictement supérieure à son homologue du second. Parallèlement, ce terme permet l'exécution d'un troisième programme. Ceci peut bien entendu être employé dans n'importe quel but, et intuitivement l'exécution du processus s'arrêtera lorsque l'un de ces deux calculs sera terminé, en donnant le résultat de celui-ci. Mais l'on peut aussi voir ce troisième argument comme étant la partie du programme qui intègre la contrainte temporelle associée à cette exécution. Il permet de n'impartir à ce programme qu'un temps d'exécution limité, ici n passages dans la boucle, et l'exécution s'arrête au bout de ce temps même si le résultat du calcul n'a pas été obtenu. La proposition 3.2.17 donne la spécification des termes qui effectuent le même travail mais sans intégrer de contrainte temporelle.

Remarques :

- La fin de l'exécution dans la troisième alternative peut surprendre. On s'attendrait plutôt à ce que cette exécution se termine sur un processus de la forme $\alpha^{0}_{k-k',0} \star t \cdot \rho$, étant donné que l'instruction $\beta_{k',l'}$ décrémente l'entier k' jusqu'au bout, avant de commencer à décrémenter l'entier l'. C'est ce fait qui nous empêche de tirer d'autres applications de la spécification associée à ce théorème, notamment en essayant de changer la manière qu'à la suite d'instruction β de décrémenter les deux entiers qui lui sont associés.
- Considérant qu'un terme réalisant une formule de la forme $A \wedge B$ réalise aussi la formule $\bot \rightarrow \bot$, on en déduit que si l'on note $(Y)\lambda r \lambda a \lambda b \lambda c (b) \Theta$ le terme donné au théorème 3.2.10, alors le terme $(Y)\lambda r \lambda a \lambda b \lambda c (a)(b) \Theta$ réalise également le théorème considéré ici pour tout $\perp\!\!\!\perp$. C'est-à-dire qu'on ne peut décider quel sera le premier de ses arguments que mettra en tête un terme réalisant ce théorème. Mais comme l'instruction $\alpha_{k,l}$ « donne deux arguments à son argument », $\beta_{k',l'}$ apparaîtra en tête dans tous les cas.

Démonstration. On prend $\perp\!\!\!\perp =< \theta \star \alpha_{k,l} \cdot \beta_{k',l'} \cdot \gamma_n \cdot \pi >$, pour une pile π quelconque.

1er cas : $k' \leqslant k$, $l' \leqslant l$ et $k'+l' \leqslant n$. On considère que θ réalise la formule $A_{\sqsubseteq}(k',l',k'+l'),\ A_{\sqsupseteq}(k',l',k'+l'+1) \to (k'+l'+1) \sqsubseteq k'+l'$. Le lemme 3.2.7 assure $\alpha_{k,l} \Vdash A_{\sqsubseteq}(k',l',k'+l')$, et le lemme 3.1.13 donne $\gamma_n \Vdash \neg^{2(k'+l')}\top$. On en déduit que $\beta_{k',l'}$ ne réalise pas la formule $A_{\sqsupseteq}(k',l',k'+l'+1)$ pour ce $\perp\!\!\!\perp$. On va démontrer par induction sur le couple (k',l'), en utilisant l'ordre lexicographique, que cette assertion donne le résultat.
Si $k' = l' = 0$, on a $\beta_{0,0} \nVdash (\top \to \perp) \vee (\top \to \perp)$. C'est-à-dire qu'il existe deux termes a et b et une pile ρ tels que $\beta_{0,0} \star a \cdot b \cdot \rho$ soit dans le fil considéré, ce qui est le résultat.
Sinon, il existe un ensemble de piles Φ, ρ appartenant à Φ, deux termes a, b réalisant respectivement les formules

$$\forall z \left\{ \forall x (A_{\sqsupseteq}(x,l',z) \to sx \neq k') \to sz \neq k'+l'+1 \right\} \to \Phi \text{ et}$$

$$\forall z \left\{ \forall y (A_{\sqsupseteq}(k',y,z) \to sz \neq l') \to sy \neq k'+l'+1 \right\} \to \Phi \text{ tels que } \beta_{k',l'} \star a \cdot b \cdot \rho$$

soit dans le fil considéré.
Si $k' = 0$ et $l' \neq 0$: ce processus se réduit en $b \star \beta^0_{0,l'} \cdot \rho$. C'est donc que $\beta^0_{0,l'}$ ne réalise pas la formule $(A_{\sqsupseteq}(0,l'-1,l') \to \perp) \to \perp$. C'est-à-dire qu'il existe un terme t réalisant la formule $A_{\sqsupseteq}(0,l'-1,l') \to \perp$ et une pile ρ_1 tels que $\beta^0_{0,l'-1} \star t \cdot \rho_1$ soit dans le fil considéré. Ce processus se réduisant en $t \star \beta_{0,l'-1} \cdot \rho_1$, on obtient que $\beta_{0,l'-1}$ ne réalise pas $A_{\sqsupseteq}0,l'-1,l'$. On obtient alors le résultat par l'hypothèse d'induction.
Si $k' \neq 0$: on obtient alors que $a \star \beta^1_{k',l'} \cdot \rho$ est dans le fil considéré. C'est donc que $\beta^1_{k',l'}$ ne réalise pas la formule $(A_{\sqsupseteq}(k'-1,l',k'+l') \to \perp) \to \perp$. C'est-à-dire qu'il existe un terme t réalisant $A_{\sqsupseteq}(k'-1,l',k'+l') \to \perp$ et une pile ρ_1 tel que $\beta^1_{k',l'} \star t \cdot \rho_1$ soit dans le fil considéré. Ce processus se réduisant en $t \star \beta_{k'-1,l'} \cdot \rho_1$, on obtient que $\beta_{k'-1,l'}$ ne réalise pas $A_{\sqsupseteq}(k'-1,l',k'+l')$, ce qui donne le résultat par l'hypothèse d'induction.

2ième cas : $k' > k$ et $k \leqslant n$. On considère que θ réalise la formule $A_{\sqsubseteq}(k+1,0,k),\ A_{\sqsupseteq}(k+1,0,k+1) \to k+1 \sqsubseteq k$. Le lemme 3.2.6 assure que $\beta_{k',l'}$ réalise $A_{\sqsupseteq}(k+1,0,k+1)$, et de plus on a $\gamma_n \Vdash \neg^{2k}\top$, c'est donc que $\alpha_{k,l}$ ne réalise pas $A_{\sqsubseteq}(k+1,0,k)$. On va démontrer par induction sur k que cette assertion donne le résultat.

Si $k = 0$, il existe un ensemble de pile Φ, un terme t réalisant $(\top \to \perp), \top \to \Phi$ et une pile $\rho \in \Phi$ tels que $\alpha_{k,l} \star t \cdot \rho$ soit dans le fil considéré. Ce processus se réduit en $t \star \alpha^1_{k,l} \cdot \alpha^0_{k,l} \cdot \rho$, et l'on obtient que $\alpha^1_{k,l}$ ne réalise pas $\top \to \perp$. C'est-à-dire qu'il existe un terme t' et une pile ρ' tels que $\alpha^1_{k,l} \star t' \cdot \rho'$ apparaît dans le fil, ce qui est le résultat recherché.

Sinon, il existe un ensemble de piles Φ, un terme t réalisant
$\left\{(A_{\sqsubseteq}(k,0,k-1)\to\bot)\to\bot\right\}, \left\{(A_{\sqsubseteq}(k+1,0,k-1)\to\bot)\to\top\right\}\to\Phi$
et une pile $\rho\in\Phi$ tels que $\alpha_{k,l}\star t\cdot\rho$ apparaisse dans le fil. Ce processus se réduisant en $t\star\alpha^1_{k,l}\cdot\alpha^0_{k,l}\cdot\rho$, on est dans l'alternative suivante :
$\alpha^1_{k,l}\nVdash(A_{\sqsubseteq}(k,0,k-1)\to\bot)\to\bot$ ou $\alpha^0_{k,l}\nVdash\top$. La deuxième éventualité étant impossible, on en déduit qu'il existe $t_1\Vdash(A_{\sqsubseteq}(k,0,k-1)\to\bot)\to\bot$ et une pile ρ_1 tels que $\alpha^1_{k,l}\star t_1\cdot\rho_1$ apparaisse dans le fil considéré. Ce processus se réduisant en $t_1\star\alpha_{k-1,l}\cdot\rho_1$, c'est donc que $\alpha_{k-1,l}\nVdash A_{\sqsubseteq}(k,0,k-1)$, ce qui donne le résultat par l'hypothèse d'induction.

3$^{\text{ième}}$ cas : $k'\leqslant k$, $l'>l$ et $k'+l\leqslant n$. On considère que θ réalise la formule $A_{\sqsubseteq}(k',l+1,k'+l),\ A_{\sqsupseteq}(k',l+1,k'+l+1)\to(k'+l+1)\sqsubseteq(k'+l)$. Les lemmes 3.2.6 et 3.1.13 assurent que $\beta_{k',l'}\Vdash A_{\sqsupseteq}(k',l+1,k'+l+1)$ et $\gamma_n\Vdash\neg^{2(k'+l)}\top$; on a donc $\alpha_{k,l}\nVdash A_{\sqsubseteq}(k',l+1,k'+l)$.
Si $k=l=0$, il existe $\Phi\subset\Pi$, un terme t réalisant $\top,(\top\to\bot)\to\Phi$ et une pile $\rho\in\Phi$ tels que $\alpha_{k,l}\star t\cdot\rho$ soit dans le fil considéré. Ce processus se réduit en $t\star\alpha^1_{0,0}\cdot\alpha^0_{0,0}\cdot\rho$, et l'on obtient que $\alpha^0_{0,0}$ ne réalise pas $\top\to\bot$. C'est-à-dire qu'il existe un terme t' et une pile ρ' tels que $\alpha^0_{0,0}\star t'\cdot\rho'$ apparaît dans le fil, ce qui est le résultat recherché.
Sinon, il existe un ensemble de piles Φ, un terme t réalisant
$\forall x\left\{\forall z(A_{\sqsubseteq}(x,l+1,z)\to sz\neq k'+l)\to sx\neq k'\right\}$,
$\forall y\left\{\forall z(A_{\sqsubseteq}(k',y,z)\to sz\neq k'+l)\to sy\neq l+1\right\}\to\Phi$ et une pile $\rho\in\Phi$ tels que $\alpha_{k,l}\star t\cdot\rho$ apparaisse dans le fil. Ce processus se réduit en $t\star\alpha^1_{k,l}\cdot\alpha^0_{k,l}\cdot\rho'$ pour une pile ρ'. C'est donc que :

- soit $\alpha^1_{k,l}\nVdash\forall x\left\{\forall z(A_{\sqsubseteq}(x,l+1,z)\to sz\neq k'+l)\to sx\neq k'\right\}$,
- soit $\alpha^0_{k,l}\nVdash\forall y\left\{\forall z(A_{\sqsubseteq}(k',y,z)\to sz\neq k'+l)\to sy\neq l+1\right\}$.

Si $k'\neq 0$, on ne peut assurer laquelle de ces deux assertions est vraie, c'est ce qui fait que les réalisateurs de ce théorème peuvent avoir des comportements plus ou moins erratiques.
Supposons d'abord être dans le premier cas. C'est donc que $k'\neq 0$ et qu'il existe une pile ρ ainsi qu'un terme t réalisant $A_{\sqsubseteq}(k'-1,l+1,k'+l-1)\to\bot$ tel que $\alpha^1_{k,l}\star t\cdot\rho$ soit dans le fil considéré. Ce processus se réduisant en $t\star\alpha_{k-1,l}\cdot\rho$, il vient que $\alpha_{k-1,l}$ ne réalise pas $A_{\sqsubseteq}(k'-1,l+1,k'+l-1)$, c'est-à-dire que l'on est l'alternative suivante :
$\alpha^1_{k-1,l}\nVdash\forall x(\forall z(A_{\sqsubseteq}(x,l+1,z)\to sz\neq k'+l-1)\to sx\neq k'-1)$
ou $\alpha^0_{k-1,l}\nVdash\forall y(\forall z(A_{\sqsubseteq}(k'-1,y,z)\to sz\neq k'+l-1)\to sy\neq l+1)$.
On est donc ramené à l'alternative précédente, où l'on a décrémenté les entiers k et k'.

Voyons maintenant ce qui se passe si l'on se trouve dans le second cas : il existe alors une pile ρ ainsi que t réalisant $\forall z(A_{\sqsubseteq}(k',l,z) \to sz \neq k'+l)$ tels que $\alpha^0_{k,l} \star t \cdot \rho$ soit dans le fil considéré. Si $l = 0$, on a là le résultat. Sinon, ce processus se réduit en $t \star \alpha_{k,l-1} \cdot \rho$, et il vient que $\alpha_{k,l-1}$ ne réalise pas la formule $A_{\sqsubseteq}(k',l,k'+l-1)$, c'est-à-dire que l'on a :
$\alpha^1_{k,l-1} \not\Vdash \forall x(\forall z(A_{\sqsubseteq}(x,l,z) \to sz \neq (k'+l-1)) \to sx \neq k')$,
ou $\alpha^0_{k,l-1} \not\Vdash \forall y(\forall z(A_{\sqsubseteq}(k',y,z) \to sz \neq k'+l-1 \to sy \neq l)$. On est donc ramené à l'alternative précédente, où l'on a décrémenté les entiers l et l'.

Donc si l'on itère ce raisonnement on sera à chaque fois obligé de décrémenter deux entiers : tantôt k et k', tantôt l et l' ; en fonction du réalisateur θ considéré. Considérant qu'on avait au départ $k' \leqslant k$ et $l' > l$, on finit par se ramener à l'une des deux situations suivantes (on regarde le premier instant où l'un des deux entiers k', l s'annule) :

i) $\alpha^1_{k-\tilde{k},0} \not\Vdash \forall x(\forall z(A_{\sqsubseteq}(x,1,z) \to sz \neq k'-\tilde{k}+1) \to sx \neq k'-\tilde{k})$, ou
$\alpha^0_{k-\tilde{k},0} \not\Vdash \forall y(\forall z(A_{\sqsubseteq}(k'-\tilde{k},y,z) \to sz \neq k'-\tilde{k}+1) \to sy \neq 1)$, avec $0 \leqslant \tilde{k} < k'$.

ii) $\alpha^1_{k-k',l-\tilde{l}} \not\Vdash \forall x(\forall z(A_{\sqsubseteq}(x,l+1-\tilde{l},z) \to sz \neq l+1-\tilde{l}) \to sx \neq 0)$, ou
$\alpha^0_{k-k',l-\tilde{l}} \not\Vdash \forall y(\forall z(A_{\sqsubseteq}(0,y,z) \to sz \neq l-\tilde{l}) \to sy \neq l+1-\tilde{l})$, avec $0 \leqslant \tilde{l} < l$.

Voyons comment conclure dans chacun des cas :

i) On va montrer que cette alternative donne le résultat par induction sur k'. Supposons que la seconde assertion soit vraie. Il existe alors une pile ρ et un terme t réalisant $\forall z(A_{\sqsubseteq}(k'-\tilde{k},0,z) \to sz \neq k'-\tilde{k}+1)$ tels que le processus $\alpha^0_{k-\tilde{k},0} \star t \cdot \rho$ apparaisse dans le fil considéré, ce qui donne le résultat. Sinon la première assertion est vraie, il existe donc une pile ρ et un terme t réalisant $\forall z(A_{\sqsubseteq}(k'-\tilde{k},1,z) \to sz \neq k'-\tilde{k}+1)$ tels que $\alpha^1_{k-\tilde{k},0} \star t \cdot \rho$ apparaisse dans le fil considéré. Ce processus se réduisant en $t \star \alpha_{k-\tilde{k}-1,0} \cdot \rho$, on obtient que $\alpha_{k-\tilde{k},0}$ ne réalise pas la formule $A_{\sqsubseteq}(k'-\tilde{k}-1,1,k'-\tilde{k})$. On obtient alors l'alternative suivante :

j) $\alpha^1_{k-\tilde{k}-1,0} \not\Vdash \forall x(\forall z(A_{\sqsubseteq}(x,1,z) \to sz \neq k'-\tilde{k}) \to sx \neq k'-\tilde{k}-1)$

ou

jj) $\alpha^0_{k-\tilde{k},0} \not\Vdash \forall y(\forall z(A_{\sqsubseteq}(k'-\tilde{k}-1,y,z) \to sz \neq k'-\tilde{k}) \to sy \neq 1)$,

ce qui donne le résultat par hypothèse d'induction.

ii) La première assertion étant toujours fausse, on en déduit qu'il existe une pile ρ et t réalisant $\forall z(A_{\sqsubseteq}(0,l-\tilde{l},z) \to sz \neq l-\tilde{l})$ tel que $\alpha^0_{k-k',l-\tilde{l}} \star t \cdot \rho$ soit dans le fil considéré. Ce processus se réduisant en $t \star \alpha_{k-k',l-\tilde{l}-1} \cdot \rho$, on en déduit que $\alpha_{k-k',l-\tilde{l}-1}$ ne réalise pas la formule $A_{\sqsubseteq}(0,l-\tilde{l},l-\tilde{l}-1)$. On se trouve dans l'alternative suivante :

$j)$ $\alpha^1_{k-k',l-\tilde{l}-1} \not\Vdash \forall x(\forall z(A_{\sqsubseteq}(x,l-\tilde{l},z) \to sz \neq l-\tilde{l}-1) \to sx \neq 0)$,

ou

$jj)$ $\alpha^0_{k-k',l-\tilde{l}-1} \not\Vdash \forall y(\forall z(A_{\sqsubseteq}(0,y,z) \to sz \neq l-\tilde{l}-1) \to sy \neq l-\tilde{l})$,

On va alors montrer que cette alternative donne le résultat par induction sur l.
La première assertion étant toujours fausse, on en déduit qu'il existe une pile ρ' et un terme t' réalisant $\forall z(A_{\sqsubseteq}(0,l-\tilde{l},z) \to sz \neq l-\tilde{l}-1)$ tel que le processus $\alpha^0_{k-k',l-\tilde{l}-1} \star t' \cdot \rho'$ apparaisse dans le fil considéré. Si $l = \tilde{l}+1$ l'exécution s'arrête et on a là le résultat. Sinon, on en déduit que $\alpha_{k-k',l-\tilde{l}-2}$ ne réalise pas la formule $A_{\sqsubseteq}(0,l-\tilde{l}-1,l-\tilde{l}-2)$; d'où l'alternative suivante :

$k)$ $\alpha^1_{k-k',l-\tilde{l}-2} \not\Vdash \forall x(\forall z(A_{\sqsubseteq}(x,l-\tilde{l}-1,z) \to sz \neq l-\tilde{l}-2) \to sx \neq 0)$,

ou

$kk)$ $\alpha^0_{k-k',l-\tilde{l}-2} \not\Vdash \forall y(\forall z(A_{\sqsubseteq}(0,y,z) \to sz \neq l-\tilde{l}-2) \to sy \neq l-\tilde{l}-1)$,

ce qui donne le résultat par l'hypothèse d'induction.

$4^{\text{ième}}$ cas : si l'on ne se trouve dans aucun des cas précédents, on est dans l'alternative suivante :

$$(n < k' \text{ et } n < k) \text{ ou } (k' \leqslant n, k' \leqslant k, n < k'+l \text{ et } n < k'+l').$$

En effet, nous allons démontrer que la négation de cette disjonction donne l'une des trois situations précédentes :

- Si $n \geqslant k'$ et $n < k'$: on a évidemment une contradiction.
- Si $n \geqslant k'$ et $k' > k$: on est dans le second cas.
- Si $n \geqslant k'$, $k' \leqslant k$ et $n \geqslant k'+l$: si de plus on a $l' > l$ on se retrouve dans le troisième cas ; si $l' \leqslant l$ on obtient $n \geqslant k'+l'$, on est alors dans le premier cas.
- Si $n \geqslant k'$, $k' \leqslant k$, $n < k'+l$ et $n \geqslant k'+l'$: si $l' > l$ on a une contradiction, et sinon on est dans le premier cas.
- Si $n \geqslant k$ et $k' > k$: on est dans le second cas.
- Si $n \geqslant k$, $k' \leqslant k$ et $n < k'$: on a évidemment une contradiction.
- Si $n \geqslant k$, $k' \leqslant k$, $n \geqslant k'$ et $n \geqslant k'+l$: Si $l' \leqslant l$ on est dans le premier cas, sinon dans le troisième.
- Si $n \geqslant k$, $k' \leqslant k$ et $n \geqslant k'$, $n < k'+l$ et $n \geqslant k'+l'$: Si $l' \leqslant l$ on est dans le premier cas, et sinon on a une contradiction.

On peut alors conclure aisément :

1. Si $n < k'$ et $n < k$: on considère que θ réalise la formule suivante : $A_{\sqsubseteq}(n+1,0,n+1)$, $A_{\sqsupseteq}(n+1,0,n+1) \to n+1 \sqsubseteq n+1$. On a que $\alpha_{k,l}$ réalise $A_{\sqsubseteq}(n+1,0,n+1)$ et $\beta_{k',l'} \Vdash A_{\sqsupseteq}(n+1,0,n+1)$, par les lemmes 3.2.7 et 3.2.6. Il vient $\gamma_n \not\Vdash \neg^{2n+1}\top$, et l'on conclut alors grâce au lemme 3.1.14.

2. Si $n \geqslant k'$, $k' \leqslant k$, $n < l + k'$ et $n < l' + k'$: on considère que θ réalise $A_{\sqsubseteq}(k', n-k'+1, n+1)$, $A_{\sqsupseteq}(k', n-k'+1, n+1) \to n+1 \sqsubseteq n+1$. Mais on a $\alpha_{k,l} \Vdash A_{\sqsubseteq}k', n-k'+1, n+1$ et $\beta_{k',l'} \Vdash A_{\sqsupseteq}k', n-k'+1, n+1$, par les lemmes 3.2.7 et 3.2.6. Il vient $\gamma_n \nVdash \neg^{2n+1}\top$, ce qui permet de conclure grâce au lemme 3.1.14.

□

3.2.3 Commutativité

On peut alors réaliser une formule exprimant que l'addition induite par la conjonction A des relations $A_{\sqsubseteq}$ et $A_{\sqsupseteq}$ sur les modèles génériques quotientés est commutative à l'aide des deux lemmes suivants.

Lemme 3.2.12.
La formule $\forall x' \forall x'' \forall x (A_{\sqsubseteq} x', x'', x \to A_{\sqsubseteq} x'', x', x)$ *est réalisée par la quasi-preuve suivante :*

$$(Y)\lambda r \lambda u \lambda v (u) \lambda a \lambda b \left((v)\lambda d (b) \lambda e (d) (r) e\right) \lambda d (a) \lambda e (d) (r) e.$$

On trouvera la preuve suivante présentée en séquents dans l'annexe A (voir page 165).

Démonstration. Soit θ le terme considéré. On va montrer par induction sur p, que quels que soient les entiers m et n, le terme u réalisant $A_{\sqsubseteq} m, n, p$, l'ensemble de piles Φ, le terme v réalisant

$$\forall x \left(\forall z (A_{\sqsubseteq} x, m, z \to sz \neq p) \to sx \neq n\right), \forall y \left(\forall z (A_{\sqsubseteq} n, y, z \to sz \neq p) \to sy \neq m\right) \to \Phi,$$

et la pile $\pi \in \Phi$, le processus $\theta \star u \cdot v \cdot \pi$ est dans $\perp\!\!\!\perp$. Celui-ci se réduisant en $u \star \lambda a \lambda b \left((v)\lambda d (b) \lambda e (d) (r) e\right) \lambda d (a) \lambda e (d) (\theta) e \cdot \pi$, il suffit (considérant la formule réalisée par u) de démontrer que $\lambda a \lambda b \left((v)\lambda d (b) \lambda e (d) (\theta) e\right) \lambda d (a) \lambda e (d) (\theta) e$ réalise

$$\forall x \left(\forall z (A_{\sqsubseteq} x, n, z \to sz \neq p) \to sx \neq m\right), \forall y \left(\forall z (A_{\sqsubseteq} m, y, z \to sz \neq p) \to sy \neq n\right) \to \{\pi\}.$$

Soient donc a et b deux termes réalisant (respectivement) les formules $\forall x (\forall z (A_{\sqsubseteq} x, n, z \to sz \neq p) \to sx \neq m)$ et $\forall y (\forall z (A_{\sqsubseteq} m, y, z \to sz \neq p) \to sy \neq n)$. On doit montrer que le processus $v \star \lambda d (b) \lambda e (d) (\theta) e \cdot \lambda d (a) \lambda e (d) (\theta) e \cdot \pi$ est dans $\perp\!\!\!\perp$. Il suffit pour cela, considérant la formule réalisée par v, de montrer

$$\lambda d (b) \lambda e (d) (\theta) e \Vdash \forall x (\forall z (A_{\sqsubseteq} x, m, z \to sz \neq p) \to sx \neq n),$$

$$\text{et } \lambda d (a) \lambda e (d) (\theta) e \Vdash \forall y (\forall z (A_{\sqsubseteq} n, y, z \to sz \neq p) \to sy \neq m).$$

On va prouver la première assertion, la seconde se démontrant de manière analogue. Si $n = 0$, il n'y a rien à démontrer. Sinon, on doit montrer que si d réalise $\forall z (A_{\sqsubseteq} (n-1, m, z) \to sz \neq p)$ et $\rho \in \Pi$, alors le processus $\lambda d (b) \lambda e (d) (\theta) e \star d \cdot \rho$

est dans $\perp\!\!\!\perp$. Mais celui-ci se réduit en $b \star \lambda e(d)(\theta)e \cdot \rho$, avec b qui réalise $\forall z(A_{\sqsubseteq}(m, n-1, z) \to sz \neq p) \to \bot$. Il suffit donc de montrer que $\lambda e(d)(\theta)e$ réalise $\forall z(A_{\sqsubseteq}(m, n-1, z) \to sz \neq p)$. Encore une fois il n'y a rien à démontrer si $p = 0$, et sinon il suffit de démontrer $d \star (\theta)e \cdot \rho' \in \perp\!\!\!\perp$ pour e réalisant $A_{\sqsubseteq}(m, n-1, p-1)$ et $\rho' \in \Pi$. Puisque $d \Vdash A_{\sqsubseteq}(n-1, m, p-1) \to \bot$, le résultat découle de l'hypothèse d'induction.

□

Lemme 3.2.13.
La formule $\forall x' \forall x'' \forall x(A_{\sqsupseteq}x', x'', x \to A_{\sqsupseteq}x'', x', x)$ *est réalisée par la quasi-preuve suivante :*

$$(Y)\lambda r \lambda u \lambda a \lambda b\,((u)\lambda c(b)\lambda d(c)\lambda e(d)(r)e)\,\lambda c(a)\lambda d(c)\lambda e(d)(r)e.$$

La preuve qui suit est présentée en séquents dans l'annexe A (voir page 168).

Démonstration. On appelle θ la quasi-preuve considérée. On va prouver par induction sur p que quels que soient les entiers m et n, le terme u réalisant $A_{\sqsupseteq}m, n, p$, l'ensemble Φ de piles, $\pi \in \Phi$, les termes a et b réalisant respectivement les formules $\forall z(\forall x(A_{\sqsupseteq}x, m, z \to sx \neq n) \to sz \neq p) \to \Phi$ et $\forall z(\forall y(A_{\sqsupseteq}n, y, z \to sy \neq m) \to sz \neq p) \to \Phi$, le processus $\theta \star u \cdot a \cdot b \cdot \pi$ est dans $\perp\!\!\!\perp$. Celui-ci se réduisant en $u \star \lambda c(b)\lambda d(c)\lambda e(d)(\theta)e \cdot \lambda c(a)\lambda d(c)\lambda e(d)(\theta)e \cdot \pi$, il suffit de prouver que :

$$\lambda c(b)\lambda d(c)\lambda e(d)(\theta)e \Vdash \forall z(\forall x(A_{\sqsupseteq}x, n, z \to sx \neq m) \to sz \neq p) \to \{\pi\},$$

$$\text{et } \lambda c(a)\lambda d(c)\lambda e(d)(\theta)e \Vdash \forall z(\forall y(A_{\sqsupseteq}m, y, z \to sy \neq n) \to sz \neq p) \to \{\pi\}.$$

Nous allons prouver la première assertion, la seconde se démontrant de manière analogue. Soit donc c un terme réalisant $\forall z(\forall x(A_{\sqsupseteq}x, n, z \to sx \neq m) \to sz \neq p)$, montrons $b \star \lambda d(c)\lambda e(d)(\theta)e \cdot \pi \in \perp\!\!\!\perp$. Considérant la formule réalisée par b, il suffit de prouver que $\lambda d(c)\lambda e(d)(\theta)e$ réalise $\forall z(\forall y(A_{\sqsupseteq}n, y, z \to sy \neq m) \to sz \neq p)$.
Si $p = 0$, il n'y a rien à démontrer. Sinon, on doit prouver que si d réalise $\forall y(A_{\sqsupseteq}(n, y, p-1) \to sy \neq m)$ et $\rho \in \Pi$, alors le processus $c \star \lambda e(d)(\theta)e \cdot \rho$ est dans $\perp\!\!\!\perp$. Mais comme c réalise $\forall x(A_{\sqsupseteq}x, n, p-1 \to sx \neq m) \to \bot$, il suffit de prouver $\lambda e(d)(\theta)e \Vdash \forall x(A_{\sqsupseteq}x, n, p-1 \to sx \neq m)$. Si $m = 0$ le résultat est trivial, et sinon on conclut grâce à l'hypothèse d'induction.

□

Théorème 3.2.14.
Soit θ un terme réalisant $\forall x'\forall x''\forall x(A_{\sqsupseteq}x', x'', x \to A_{\sqsupseteq}x'', x', x)$, et θ' un terme réalisant $\forall x'\forall x''\forall x(A_{\sqsubseteq}x', x'', x \to A_{\sqsubseteq}x'', x', x)$.
Alors la formule $\forall x\forall x'\forall x''\forall z(A_{\sqsubseteq}x', x'', x\ ,\ A_{\sqsupseteq}x'', x', z \to z \sqsubseteq x)$ est réalisée par chacune des deux quasi-preuves suivantes :

i) $(Y)\lambda r\lambda a\lambda b\lambda c\left((\theta)b\lambda x(x)\lambda y((a)\dot{1})\lambda z(c)(r)zy\right)\lambda x(x)\lambda y((a)\dot{0})\lambda z(c)(r)zy$

ii) $(Y)\lambda r\lambda a\lambda b\lambda c\left((b)\lambda x(x)\lambda y((\theta')a\dot{1})\lambda z(c)(r)zy\right)\lambda x(x)\lambda y((\theta')a\dot{0})\lambda z(c)(r)zy.$

Démonstration. La preuve est analogue à celle du théorème 3.2.10, et utilise les lemmes 3.2.12 et 3.2.13. □

On en déduit une quasi-preuve réalisant le théorème exprimant la commutativité de l'addition définie sur l'espace quotient :

$$\forall x\forall x'\forall x''\forall z(A(x', x'', x)\ ,\ A(x'', x', z) \to z \equiv x),$$

en utilisant deux fois le résultat ci-dessus.

La formule $\forall x\forall y(x + y = y + x)$ est réalisée par $\lambda x\ x$, et sa spécification indique que toute quasi-preuve la réalisant se comporte comme ce terme. La méthode développée dans ce chapitre est une manière d'associer un contenu opérationnel non-trivial aux théorèmes de cette forme.

On peut maintenant donner la spécification associée au théorème 3.2.14, qui est analogue à celle du théorème 3.2.10.

Théorème 3.2.15.
Soit θ réalisant la formule $\forall x\forall x'\forall x''\forall z(A_{\sqsubseteq}(x', x'', x),\ A_{\sqsupseteq}(x'', x', z) \to z \sqsubseteq x)$
Alors, pour toute pile π, l'exécution de $\theta \star \alpha_{k,l} \cdot \beta_{k',l'} \cdot \gamma_n \cdot \pi$ se termine sur un processus de la forme :

- $\beta^{\tau}_{0,0} \star t \cdot \rho$ *avec τ égal à 0 ou 1, si $k' \leqslant l$, $l' \leqslant k$ et $k' + l' \leqslant n$,*
- $\alpha^{0}_{k,0} \star t \cdot \rho$ *si $k' > l$ et $l \leqslant n$,*
- $\alpha^{1}_{0,\bar{l}} \star t \cdot \rho$ *si $k' \leqslant l$, $l' > k$ et $k' + k \leqslant n$, où $\bar{l}$ est compris entre $l - k'$ et l,*
- $\gamma_0 \star t \cdot \rho$ *sinon.*

Démonstration. Analogue à celle du théorème 3.2.10. □

Une méthode alternative

On peut à l'aide des lemmes 3.2.12 et 3.2.13 réaliser une autre formule exprimant elle aussi la commutativité de l'addition induite :

$$\forall x'\forall x''\forall x(Ax', x'', x \to Ax'', x', x)$$

La spécification de celle-ci s'obtient aisément à partir de celles des théorèmes réalisés aux lemmes 3.2.12 et 3.2.13.

On donne ci-dessous la spécification associée au lemme 3.2.13. On utilise pour cela la suite de processus α, mais afin d'obtenir le résultat le plus clair possible on ajoute des nouvelles instructions $\alpha_{-1,\tilde{l}}$ et $\alpha_{\tilde{k},-1}$ quels que soient les entiers $\tilde{k}$ et $\tilde{l}$, ainsi que les règles d'exécution suivante :

$$\alpha^1_{0,\tilde{l}} \star t \cdot \pi \succ t \star \alpha_{-1,\tilde{l}} \cdot \pi$$

$$\alpha^0_{\tilde{k},0} \star t \cdot \pi \succ t \star \alpha_{\tilde{k},-1} \cdot \pi$$

Lemme 3.2.16.
Soit $p \in \Lambda_c \star \Pi$ et $\perp\!\!\!\perp =< p >$. Avec les hypothèses supplémentaires ci-dessus, si $p > k + l$, on a :

1. $\alpha^1_{k,l} \Vdash \forall z(\forall x(A_{\sqsupseteq} x, l, z \to sx \neq k) \to sz \neq p) \to \bot$
2. $\alpha^0_{k,l} \Vdash \forall z(\forall y(A_{\sqsupseteq} k, y, z \to sy \neq l) \to sz \neq p) \to \bot$

Démonstration. On va montrer le résultat suivant par récurrence dite « descendante » sur $\tilde{k} + \tilde{l}$:
Quels que soient $\tilde{l}$ et $\tilde{k}$ entiers tels que $0 \leqslant \tilde{l} \leqslant l$ et $0 \leqslant \tilde{k} \leqslant k$ on a :
$\alpha^1_{k-\tilde{k},l-\tilde{l}} \Vdash \forall z(\forall x(A_{\sqsupseteq}(x, l-\tilde{l}, z) \to sx \neq k-\tilde{k}) \to sz \neq p-\tilde{k}-\tilde{l}) \to \bot$
et $\alpha^0_{k-\tilde{k},l-\tilde{l}} \Vdash \forall z(\forall y(A_{\sqsupseteq}(k-\tilde{k}, y, z) \to sy \neq l-\tilde{l}) \to sz \neq (p-\tilde{k}-\tilde{l})) \to \bot$.

Si $\tilde{k} + \tilde{l} = k + l$: on doit montrer que $\alpha^1_{0,0}$ et $\alpha^0_{0,0}$ réalisent $(\top \to \bot) \to \bot$, ce qui découle de l'ajout d'instructions supplémentaires.
Si $\tilde{k} + \tilde{l} < k + l$:

1. Montrons $\alpha^1_{k-\tilde{k},l-\tilde{l}} \star t \cdot \pi \in \perp\!\!\!\perp$ si π est une pile et t un terme réalisant $\forall z(\forall x(A_{\sqsupseteq}(x, l-\tilde{l}, z) \to sx \neq k-\tilde{k}) \to sz \neq (p-\tilde{k}-\tilde{l}))$. Ce processus se réduit en $t \star \alpha_{k-\tilde{k}-1,l-\tilde{l}} \cdot \pi$. Il suffit donc de montrer que $\alpha_{k-\tilde{k}-1,l-\tilde{l}}$ réalise $\forall x(A_{\sqsupseteq}(x, l-\tilde{l}, p-\tilde{k}-\tilde{l}-1) \to sx \neq k-\tilde{k})$. Si $\tilde{k} = k$, il n'y a rien à montrer. Sinon, on prend une pile ρ, un terme t' réalisant $A_{\sqsupseteq}(k-\tilde{k}, l-\tilde{l}, p-\tilde{k}-\tilde{l}-1)$ et il suffit de montrer que $\alpha_{k-\tilde{k}-1,l-\tilde{l}} \star t' \cdot \rho$ est dans $\perp\!\!\!\perp$. Ce processus se réduisant en $t' \star \alpha^1_{k-\tilde{k}-1,l-\tilde{l}} \cdot \alpha^0_{k-\tilde{k}-1,l-\tilde{l}} \cdot \rho$, il suffit de prouver que $\alpha^1_{k-\tilde{k}-1,l-\tilde{l}}$ et $\alpha^0_{k-\tilde{k}-1,l-\tilde{l}}$ réalisent respectivement les formules suivantes :
   $\forall z(\forall x(A_{\sqsupseteq}(x, l-\tilde{l}, z) \to sx \neq (k-\tilde{k}-1) \to sz \neq (p-\tilde{k}-\tilde{l}-1)) \to \bot$
   et $\forall z(\forall y(A_{\sqsupseteq}(k-\tilde{k}-1, y, z) \to sy \neq (l-\tilde{l}) \to sz \neq (p-\tilde{k}-\tilde{l}-1)) \to \bot$, ce qui est vrai par hypothèse de récurrence.
2. On raisonne de même pour la seconde proposition.

□

On peut enfin donner la spécification associée au lemme 3.2.13.

Théorème 3.2.17.
Soit θ un terme réalisant le théorème $\forall x' \forall x'' \forall x (A_{\sqsupseteq} x', x'', x \to A_{\sqsupseteq} x'', x', x)$.
Alors quelle que soit la pile π, l'exécution de $\theta \star \beta_{k',l'} \cdot \alpha^1_{k,l} \cdot \alpha^0_{k,l} \cdot \pi$ se termine sur un processus de la forme :

- *$\beta^\tau_{0,0} \star t \cdot \rho$ si $k' \leqslant l$ et $l' \leqslant k$, avec τ égal à 0 ou à 1,*
- *$\alpha_{\tilde{k},-1} \star t \cdot \rho$ avec $0 \leqslant \tilde{k} \leqslant k$, si $k' > l$,*
- *$\alpha_{-1,\tilde{l}} \star t \cdot \rho$ avec $0 \leqslant \tilde{l} \leqslant l$ sinon, c'est-à-dire si $k' \leqslant l$ et $l' > k$.*

Interprétation : Cette spécification est la même que la proposition 3.2.11, à ceci près que les termes obtenus ici ne permettent pas d'intégrer une contrainte temporelle.

Démonstration. Soit $\pi \in \Pi$ et $\perp\!\!\!\perp =< \theta \star \beta_{k',l'} \cdot \alpha^1_{k,l} \cdot \alpha^0_{k,l} \cdot \pi >$.
1^{er} cas : si $k' \leqslant l$ et $l' \leqslant k$. on a $\theta \Vdash A_{\sqsupseteq}(l,k,k+l+1) \to A_{\sqsupseteq}(k,l,k+l+1)$. Le lemme 3.2.16 assure que :
$\alpha^1_{k,l} \Vdash \forall z(\forall x(A_{\sqsupseteq}(x,l,z) \to sx \neq k) \to sz \neq (k+l+1)) \to \perp$
et $\alpha^0_{k,l} \Vdash \forall z(\forall y(A_{\sqsupseteq}(k,y,z) \to sy \neq l) \to sz \neq (k+l+1)) \to \perp$. On en déduit $\beta_{k',l'} \nVdash A_{\sqsupseteq}(l,k,k+l+1)$. C'est-à-dire qu'il existe une pile ρ, deux termes t_1 et t_0 réalisant respectivement
$\forall z(\forall x(A_{\sqsupseteq}(x,k,z) \to sx \neq l) \to sz \neq (k+l+1)) \to \perp$
et $\forall z(\forall y(A_{\sqsupseteq}(l,y,z) \to sy \neq k) \to sz \neq (k+l+1)) \to \perp$ tels que
$\beta_{k',l'} \star t_1 \cdot t_0 \cdot \rho$ apparaisse dans le fil considéré.

- Si $k' = l' = 0$: ce processus se réduit par exemple en $t_1 \star \beta^1_{0,0} \cdot \rho$ (on traiterait l'autre cas de la même façon), ce qui assure que $\beta^1_{0,0}$ ne réalise pas $\forall z(\forall x(A_{\sqsupseteq}(x,k,z) \to sx \neq l) \to sz \neq (k+l+1))$. On en déduit qu'il existe un terme t réalisant $\forall x(A_{\sqsupseteq}(x,k,k+l) \to sx \neq l)$ et une pile ρ' tels que le processus $\beta^1_{0,0} \star t \cdot \rho'$ apparaisse dans le fil considéré, ce qui est le résultat.
- Si $k' = 0$ et $l' \neq 0$: ce processus se réduit en $t_0 \star \beta^0_{0,l'} \cdot \rho$, ce qui assure $\beta^0_{0,l'} \nVdash \forall z(\forall y(A_{\sqsupseteq}(l,y,z) \to sy \neq k) \to sz \neq (k+l+1))$, et l'on en déduit qu'il existe t réalisant $\forall y(A_{\sqsupseteq}(l,y,k+l) \to sy \neq k)$ et une pile ρ' tels que le processus $\beta^0_{0,l'} \star t \cdot \rho'$ apparaisse dans le fil considéré. Ce processus se réduit en $t \star \beta_{0,l'-1} \cdot \rho'$, ce qui assure $\beta_{0,l'-1} \nVdash A_{\sqsupseteq}(l,k-1,k+l)$: on se ramène au cas précédent en raisonnant par induction sur l'.
- Si $k' \neq 0$: ce processus se réduit en $t_1 \star \beta^1_{k',l'} \cdot \rho$, ce qui donne $\beta^1_{k',l'} \nVdash \forall z(\forall x(A_{\sqsupseteq}(x,k,z) \to sx \neq l) \to sz \neq (k+l+1))$, et l'on en déduit qu'il existe un terme t réalisant $\forall x(A_{\sqsupseteq}(x,k,k+l) \to sx \neq l)$ et une pile ρ' tels que le processus $\beta^1_{k',l'} \star t \cdot \rho'$ apparaisse dans le fil considéré. Ce processus se réduit en $t \star \beta_{k'-1,l'} \cdot \rho'$, et donc $\beta_{k'-1,l'} \nVdash A_{\sqsupseteq}(l-1,k,k+l)$; on se ramène au cas précédent en raisonnant par induction sur k'.

2ième cas : si $k' > l$. On a $\theta \Vdash A_\sqsupseteq(l{+}1,0,l{+}1) \to A_\sqsupseteq(0,l{+}1,l{+}1)$. Mais le lemme 3.2.6 donne $\beta_{k',l'} \Vdash A_\sqsupseteq(l{+}1,0,l{+}1)$, c'est donc que :

- soit $\alpha^1_{k,l} \nVdash \forall z(\forall x(A_\sqsupseteq(x,l{+}1,z) \to sx \neq 0) \to sz \neq (l{+}1)) \to \bot$,
- soit $\alpha^0_{k,l} \nVdash \forall z(\forall y(A_\sqsupseteq(0,y,z) \to sy \neq (l{+}1)) \to sz \neq (l{+}1)) \to \bot$.

Montrons déjà que la première proposition ne peut être vraie : sinon, il existerait un terme t réalisant $\forall z(\forall x(A_\sqsupseteq(x,l{+}1,z) \to sx \neq 0) \to sz \neq l{+}1)$ et une pile ρ tels que $\alpha^1_{k,l} \star t \cdot \rho'$ apparaisse dans le fil considéré. Ce processus se réduisant en $t \star \alpha_{k-1,l} \cdot \rho$, on en déduirait $\alpha_{k-1,l} \nVdash \forall x(A_\sqsupseteq(x,l{+}1,l \to sx \neq 0)$, ce qui est absurde.
Montrons alors par induction sur l que la seconde proposition permet de conclure. Il existe t réalisant $\forall z(\forall y(A_\sqsupseteq(0,y,z) \to sy \neq l{+}1) \to sz \neq l{+}1)$ et une pile ρ tels que $\alpha^0_{k,l} \star t \cdot \rho'$ appartienne au fil considéré. Ce processus se réduisant en $t \star \alpha_{k,l-1} \cdot \rho$, on en déduit $\alpha_{k,l-1} \nVdash A_\sqsupseteq 0,l,l \to \bot$. Il existe donc un terme t' réalisant $A_\sqsupseteq(0,l,l)$ et une pile ρ' tels que $\alpha_{k,l-1} \star t' \cdot \rho'$ apparaisse dans le fil considéré.
Si $l = 0$, ceci est le résultat.
Sinon, ce processus se réduit en $t' \star \alpha^1_{k,l-1} \cdot \alpha^0_{k,l-1} \cdot \rho'$, c'est-à-dire que :
$\alpha^1_{k,l-1} \nVdash \forall z(\forall x(A_\sqsupseteq(x,l,z) \to sx \neq 0) \to sz \neq l) \to \bot$
ou $\alpha^0_{k,l-1} \nVdash \forall z(\forall y(A_\sqsupseteq(0,y,z) \to sy \neq l) \to sz \neq l) \to \bot$. La première proposition étant absurde, on peut alors conclure grâce à l'hypothèse d'induction.

3ième cas : si $k' \leqslant l$ et $l' > k$. On considère que $\theta \Vdash A_\sqsupseteq(k',l',k'{+}l') \to A_\sqsupseteq(l',k',k'{+}l')$. Le lemme 3.2.6 assurant que $\beta_{k',l'} \Vdash A_\sqsupseteq(k',l',k'{+}l')$, c'est donc que :

- soit $\alpha^1_{k,l} \nVdash \forall z(\forall x(A_\sqsupseteq(x,k',z) \to sx \neq l') \to sz \neq (k'{+}l')) \to \bot$,
- soit $\alpha^0_{k,l} \nVdash \forall z(\forall y(A_\sqsupseteq(l',y,z) \to sy \neq k') \to sz \neq (k'{+}l')) \to \bot$.

Supposons être dans le premier cas, il existe alors un terme t réalisant $\forall z(\forall x(A_\sqsupseteq(x,k',z) \to sx \neq l') \to sz \neq (k'{+}l'))$ et une pile ρ tels que $\alpha^1_{k,l} \star t \cdot \rho$ apparaisse dans le fil considéré. Ce processus se réduisant en $t \star \alpha_{k-1,l} \cdot \rho$, on en déduit $\alpha_{k-1,l} \nVdash A_\sqsupseteq(l'{-}1,k',k'{+}l'{-}1) \to \bot$. Il existe donc un terme t' réalisant $A_\sqsupseteq(l'{-}1,k',k'{+}l'{-}1)$ et une pile ρ' tels que $\alpha_{k-1,l} \star t' \cdot \rho'$ apparaisse dans le fil considéré. Si $k = 0$ on a là le résultat. Sinon, ce processus se réduit en $t' \star \alpha^1_{k-1,l} \cdot \alpha^0_{k-1,l} \cdot \rho'$, c'est-à-dire que
$\alpha^1_{k-1,l} \nVdash \forall z(\forall x(A_\sqsupseteq(x,k',z) \to sx \neq (l'{-}1)) \to sz \neq (k'{+}l'{-}1)) \to \bot$
ou $\alpha^0_{k-1,l} \nVdash \forall z(\forall y(A_\sqsupseteq(l'{-}1,y,z) \to sy \neq k') \to sz \neq (k'{+}l'{-}1)) \to \bot$. On est donc ramené à l'alternative précédente où l'on a décrémenté les entiers k et l'.
Supposons maintenant être dans le second cas, il existe alors un terme t réalisant $\forall z(\forall y(A_\sqsupseteq(l',y,z) \to sy \neq k') \to sz \neq (k'{+}l'))$ et une pile ρ tels que $\alpha^0_{k,l} \star t \cdot \rho$ apparaisse dans le fil considéré. Ce processus se réduisant en $t \star \alpha_{k,l-1} \cdot \rho$, il vient $\alpha_{k,l-1} \nVdash \forall y(A_\sqsupseteq(l',y,k'{+}l'{-}1) \to sy \neq k')$. Ceci assure que $k' \neq 0$, et il existe alors une pile ρ' et t' réalisant $A_\sqsupseteq(l',k'{-}1,k'{+}l'{-}1)$ tels que $\alpha_{k,l-1} \star t' \cdot \rho'$ apparaisse dans le fil considéré. Or on a $l \geqslant k' > 0$, donc ce processus se réduit en $t' \star \alpha^1_{k,l-1} \cdot \alpha^0_{k-1,l} \cdot \rho'$, et l'on en déduit que :

$\alpha^1_{k,l-1} \not\Vdash \forall z(\forall x(A_{\sqsupseteq}(x,k'-1,z) \to sx \neq l') \to sz \neq (k'+l'-1)) \to \bot$ ou
$\alpha^0_{k,l-1} \not\Vdash \forall z(\forall y(A_{\sqsupseteq}(l',y,z \to sy \neq (k'-1)) \to sz \neq (k'+l'-1)) \to \bot$.
On est donc encore une fois ramené à l'alternative précédente, où l'on a décrémenté les entiers k' et l.
Considérant qu'on avait au départ $k' \leqslant l$ et $l' > k$, on finit en itérant ce raisonnement par se ramener à l'une des deux situations suivantes (on regarde le premier instant où l'un des deux entiers k', k s'annule) :

i) $\alpha^1_{k-\tilde{k},l-k'} \not\Vdash \forall z(\forall x(A_{\sqsupseteq}(x,0,z) \to sx \neq l'-\tilde{k}) \to sz \neq l'-\tilde{k}) \to \bot$ ou
$\alpha^0_{k-\tilde{k},l-k'} \not\Vdash \forall z(\forall y(A_{\sqsupseteq}(l'-\tilde{k},y,z) \to sy \neq 0) \to sz \neq l'-\tilde{k}) \to \bot$, avec $0 \leqslant \tilde{k} < k$.

ii) $\alpha^1_{0,l-\tilde{l}} \not\Vdash \forall z(\forall x(A_{\sqsupseteq}(x,k'-\tilde{l},z) \to sx \neq l'-k) \to sz \neq k'+l'-k-\tilde{l}) \to \bot$ ou
$\alpha^0_{0,l-\tilde{l}} \not\Vdash \forall z(\forall y(A_{\sqsupseteq}(l'-k,y,z) \to sy \neq k'-\tilde{l}) \to sz \neq k'+l'-k-\tilde{l}) \to \bot$, avec $0 \leqslant \tilde{l} < k'$.

Voyons alors comment conclure dans chacun des cas :

i) Montrons d'abord que la deuxième proposition est absurde. Sinon, il existerait une pile ρ_2 et un terme t_2 réalisant la formule $\forall z(\forall y(A_{\sqsupseteq}(l'-\tilde{k},y,z) \to sy \neq 0) \to sz \neq l'-\tilde{k})$ tels que $\alpha^0_{k-\tilde{k},l-k'} \star t_2 \cdot \rho_2$ soit dans le fil considéré. Ce processus se réduisant en $t_2 \star \alpha_{k-\tilde{k},l-k'-1} \cdot \rho_2$, ceci assurerait que $\alpha_{k-\tilde{k},l-k'-1}$ ne réalise pas $\forall y(A_{\sqsupseteq}(l'-\tilde{k},y,l'-\tilde{k}-1) \to sy \neq 0)$, car $l'-\tilde{k} > 0$. Cette dernière assertion étant absurde, on en déduit que l'on se trouve fatalement dans le premier cas.
Montrons maintenant que la proposition
$\alpha^1_{k-\tilde{k},l-k'} \not\Vdash \forall z(\forall x(A_{\sqsupseteq}(x,0,z) \to sx \neq l'-\tilde{k}) \to sz \neq l'-\tilde{k}) \to \bot$ permet de conclure, en raisonnant par induction sur $k-\tilde{k}$. Il existe donc t_2 réalisant $\forall z(\forall x(A_{\sqsupseteq}(x,0,z) \to sx \neq l'-\tilde{k}) \to sz \neq l'-\tilde{k})$ et une pile ρ_2 tels que $\alpha^1_{k-\tilde{k},l-k'} \star t_2 \cdot \rho_2$ soit dans le fil considéré. Ce processus se réduit en
$t_2 \star \alpha_{k-\tilde{k}-1,l-k'} \cdot \rho_2$, ce qui assure que $\alpha_{k-\tilde{k}-1,l-k'}$ ne réalise pas la formule $\forall x(A_{\sqsupseteq}(x,0,l'-\tilde{k}-1) \to sx \neq l'-\tilde{k})$, puisque $l' > k > \tilde{k}$. On en déduit qu'il existe une pile ρ_3 ainsi qu'un terme t_3 réalisant $A_{\sqsupseteq}(l'-\tilde{k}-1,0,l'-\tilde{k}-1)$ tels que $\alpha_{k-\tilde{k}-1,l-k'} \star t_3 \cdot \rho_3$ apparaisse dans le fil considéré.
Si $k-\tilde{k}=0$, on a là le résultat.
Sinon, ce processus se réduisant en $t_3 \star \alpha^1_{k-\tilde{k}-1,l-k'} \cdot \alpha^0_{k-\tilde{k}-1,l-k'} \cdot \rho_3$, on en déduit que $\alpha^1_{k-\tilde{k}-1,l-k'}$ ne réalise pas la formule $\forall z(\forall x(A_{\sqsupseteq}(x,0,z) \to sx \neq l'-\tilde{k}-1) \to sz \neq l'-\tilde{k}-1) \to \bot$, ce qui permet de conclure en utilisant l'hypothèse d'induction.

ii) Supposons que la première proposition soit vraie. Il existe alors t_2 réalisant $\forall z(\forall x(A_{\sqsupseteq}(x,k'-\tilde{l},z) \to sx \neq l'-k) \to sz \neq k'+l'-k-\tilde{l})$ et une pile ρ_2 tels que $\alpha^1_{0,l-\tilde{l}} \star t_2 \cdot \rho_2$ soit dans le fil considéré. Ce processus se réduisant en $t_2 \star \alpha_{-1,l-\tilde{l}} \cdot \rho_2$, ceci assure que $\alpha_{-1,l-\tilde{l}}$ ne réalise pas

$\forall x(A_{\sqsupseteq}(x,k'-\tilde{l},k'+l'-k-\tilde{l}-1) \to sx \neq l'-k)$, car $\tilde{l} < k'$ et $l' > k$. Il existe alors t_3 réalisant $A_{\sqsupseteq}(l'-k-1,k'-\tilde{l},k'+l'-k-\tilde{l}-1) \to \bot$ et une pile ρ_3 tels que $\alpha_{-1,l-\tilde{l}} \star t_3 \cdot \rho_3$ apparaisse dans le fil considéré, ce qui est le résultat.
Si la seconde proposition est vraie, il existerait un terme t_2 réalisant $\forall z(\forall y(A_{\sqsupseteq}(l'-k,y,z) \to sy \neq k'-\tilde{l}) \to sz \neq (k'+l'-k-\tilde{l}))$ et une pile ρ_2 tels que $\alpha^0_{0,l-\tilde{l}} \star t_2 \cdot \rho_2$ soit dans le fil considéré. Ce processus se réduit en $t_2 \star \alpha_{0,l-\tilde{l}-1} \cdot \rho_2$, ce qui assure que $\alpha_{0,l-\tilde{l}-1}$ ne réalise pas $\forall y(A_{\sqsupseteq}(l'-k,y,k'+l'-k-\tilde{l}-1) \to sy \neq k'-\tilde{l})$, car $\tilde{l} < k'$ et $l' > k$. Il existe alors t_3 réalisant $A_{\sqsupseteq}(l'-k,k'-\tilde{l}-1,k'+l'-k-\tilde{l}-1) \to \bot$ et une pile ρ_3 tels que $\alpha_{0,l-\tilde{l}-1} \star t_3 \cdot \rho_3$ apparaisse dans le fil considéré. On se ramène alors à l'alternative suivante, avec $0 \leqslant \tilde{l} < k'$:
j) $\alpha^1_{0,l-\tilde{l}-1}$ne réalise pas
$\forall z(\forall x(A_{\sqsupseteq}(x,k'-\tilde{l}-1,z) \to sx \neq l'-k) \to sz \neq (k'+l'-k-\tilde{l}-1)) \to \bot$
ou
jj) $\alpha^0_{0,l-\tilde{l}-1}$ ne réalise pas
$\forall z(\forall y(A_{\sqsupseteq}(l'-k,y,z) \to sy \neq (k'-\tilde{l}-1)) \to sz \neq (k'+l'-k-\tilde{l}-1)) \to \bot$

On va démontrer par induction sur $k'-\tilde{l}-1$ que cette assertion permet de conclure.
Si $k'-\tilde{l}-1 = 0$, montrons que la seconde proposition ne peut être vérifiée, ce qui permet alors de conclure en raisonnant comme ci-dessus. Sinon, puisque $l' > k$, il existerait une pile ρ_4 et un terme t_4 réalisant $\forall y(A_{\sqsupseteq}(l'-k,y,l'-k-1) \to sy \neq 0) \to \bot$ tels que $t_4 \star \alpha^0_{0,l-\tilde{l}-1} \cdot \rho_4$ soit dans le fil considéré. Ceci impliquerait que $\alpha^0_{0,l-\tilde{l}-1}$ ne réalise pas la formule $\forall y(A_{\sqsupseteq}(l'-k,y,l'-k-1) \to sy \neq 0)$, ce qui est absurde.
Supposons maintenant $k'-\tilde{l}-1 > 0$. Encore une fois, si la première proposition est vraie, on démontre facilement le résultat. Sinon, il existe une pile ρ_4 et un terme t_4 réalisant $\forall y(A_{\sqsupseteq}(l'-k,y,k'+l'-k-\tilde{l}-2) \to sy \neq (k'-\tilde{l}-1)) \to \bot$ tels que le processus $t_4 \star \alpha^0_{0,l-\tilde{l}-1} \cdot \rho_4$ soit dans le fil considéré. On en déduit que $\alpha^0_{0,l-\tilde{l}-1}$ ne réalise pas $\forall y(A_{\sqsupseteq}(l'-k,y,k'+l'-k-\tilde{l}-2) \to sy \neq (k'-\tilde{l}-1))$. C'est-à-dire qu'il existe t_5 réalisant $A_{\sqsupseteq}(l'-k,k'-\tilde{l}-2,k'+l'-k-\tilde{l}-2)$ et une pile ρ_5 tels que $\alpha^0_{0,l-\tilde{l}-1} \star t_5 \cdot \rho_5$. Ce processus se réduisant en $t_5 \star \alpha_{0,l-\tilde{l}-2} \cdot \rho_5$, on est donc dans l'alternative suivante :
k) $\alpha^1_{0,l-\tilde{l}-2}$ ne réalise pas
$\forall z(\forall x(A_{\sqsupseteq}(x,k'-\tilde{l}-2,z) \to sx \neq (l'-k)) \to sz \neq (k'+l'-k-\tilde{l}-2)) \to \bot$
ou
kk) $\alpha^0_{0,l-\tilde{l}-2}$ ne réalise pas
$\forall z(\forall y(A_{\sqsupseteq}(l'-k,y,z) \to sy \neq (k'-\tilde{l}-2)) \to sz \neq (k'+l'-k-\tilde{l}-2)) \to \bot$

ce qui permet de conclure grâce à l'hypothèse d'induction.

□

Chapitre 4

Jeux et spécification de protocoles réseaux

Nous allons dans ce chapitre développer une correspondance introduite dans [28] entre d'une part des jeux définis sur les formules du premier ordre, et d'autre part la spécification des protocoles de la couche transport des réseaux. Celle-ci donne un éclairage nouveau sur ces protocoles, et permet d'en donner une spécification, ce qui était difficilement réalisable avec leur description par des automates. Par ailleurs, on verra que la robustesse d'un protocole correspond à la validité de la formule associée, ce qui permet d'espérer pouvoir écrire de nouveaux protocoles grâce à cette méthode.

4.1 Notions de protocole de transport

Nous allons dans cette partie décrire brièvement le protocole TCP (pour *Transmission Control Protocol*), tel qu'il est détaillé dans [32]. Il s'agit de l'un des principaux protocoles de la couche transport du modèle TCP/IP, dont le rôle est de permettre aux applications de communiquer de manière sûre, en intégrant un contrôle de livraison. TCP permet donc aux routeurs de s'affranchir de ce problème et de ne s'occuper que de l'acheminement des données à travers le réseau. Pour assurer cette fiabilité, il faut ajouter aux paquets de données un *en-tête* qui va permettre de synchroniser les transmissions et d'assurer leur réception.

4.1.1 Le principe de l'acquittement

Le protocole TCP possède un système d'accusé de réception permettant au *client* (c'est-à-dire la machine qui envoie des requêtes) et au *serveur* (la machine qui est censée répondre aux requêtes du client) de s'assurer de la bonne réception mutuelle des données.
Lors de l'émission d'un message contenant des données, le serveur associe un numéro d'ordre ou *numéro de séquence* aux données considérées, qu'il introduit dans l'en-tête du message. A la réception d'un tel message, le client retourne

un message accompagné du numéro de séquence du message reçu : ce message est appelé l'*acquittement* des données contenus dans le message initial.
Lorsque le serveur reçoit un acquittement, il sait grâce au numéro de séquence que le récepteur a bien reçu le message qu'il lui avait envoyé. Il peut ensuite envoyer d'autres données s'il le désire, mais en utilisant un autre numéro de séquence, ou stopper l'émission (Cf. la figure 4.1).

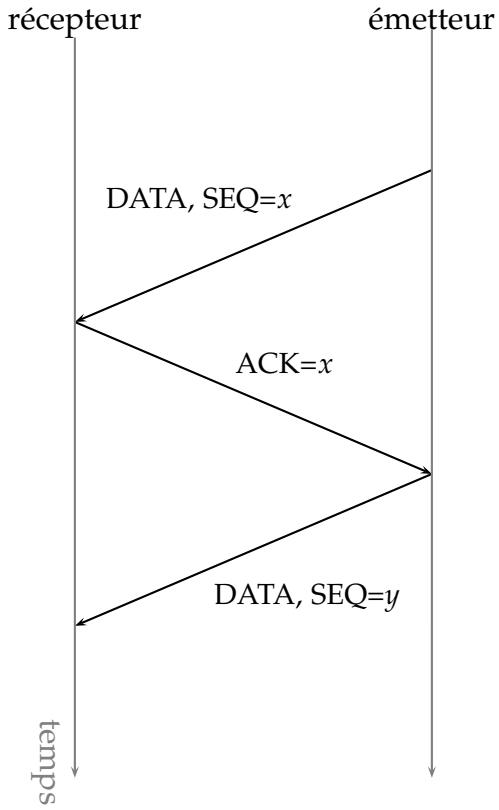


FIG. 4.1 – Expédition puis acquittement d'un message.

L'émetteur déclenche par ailleurs une minuterie à l'envoi de chaque message. S'il n'a pas reçu d'acquittement au bout d'un temps appelé *timeout*, il considère que le message n'a pas été reçu par le client et réexpédie les données (Cf. la figure 4.2).

Il se peut néanmoins que le message était arrivé à destination, mais que l'accusé de réception envoyé alors par le client a été perdu. Celui-ci pourra alors recevoir une deuxième fois les données concernées. Mais le numéro de séquence lui permet d'identifier les doublons : il doit alors simplement acquitter les données (pour que le serveur soit enfin informé que celles-ci ont été reçus, et qu'il doit envoyer les suivantes) sans les prendre en compte.

Le point central de ce principe est que le client envoie au plus un acquittement par message reçu, et que le serveur n'accuse pas réception des acquittements (c'est le paquet de données suivant qui joue ce rôle).

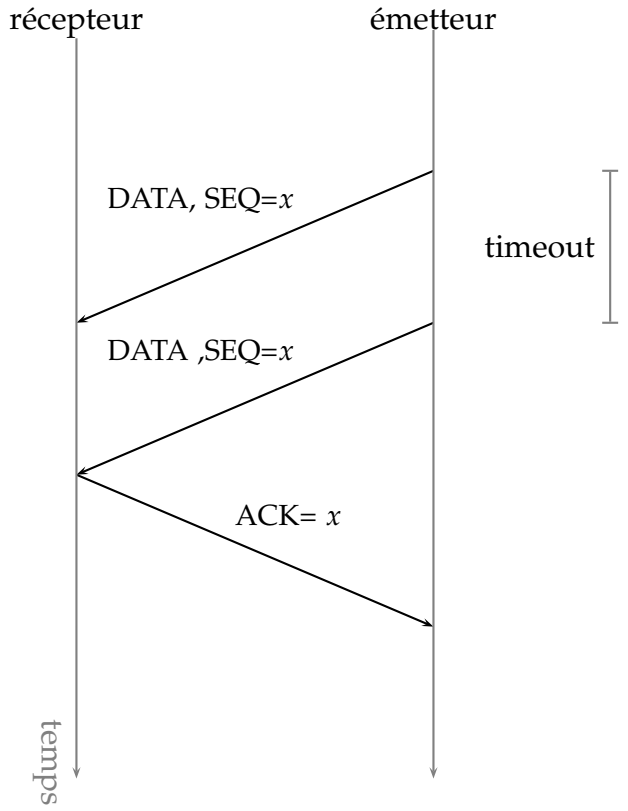


FIG. 4.2 – Expédition, réexpédition puis acquittement d'un message.

4.1.2 La procédure de connexion

Le protocole TCP n'empêche pas deux hôtes d'ouvrir et de fermer plusieurs fois une connexion, ce qui soulève le problème de la reconnaissance par chacun d'eux des messages provenant d'une instance antérieure de la connexion ou *session*.
Il faut pour cela disjoindre les numéros de séquence utilisés dans des sessions différentes, afin d'éviter que lorsqu'un paquet de données provenant d'une ancienne session (qui se trouverait encore sur le réseau) arrive au client, celui-ci ne considère qu'il s'agit de données répondant à sa dernière requête. L'initialisation de la connexion correspond donc à l'envoi par le serveur du premier numéro de séquence qu'il utilisera, celui-ci convenant de n'utiliser par la suite que des numéros de séquence strictement supérieurs au numéro initial.
Afin de permettre à chacune des parties d'amorcer la procédure d'initialisation, et d'éviter que celle-ci soit perturbée par l'arrivée inopinée de messages ayant trait à une tentative de connexion caduc (ce qui pourrait aboutir à une mésentente quand au numéro initial choisi), on utilise un deuxième numéro de

séquence. Celui-ci est initialisé par le client.
L'initialisation de la connexion correspond à l'échange des numéros de séquence initiaux choisis par chacun des hôtes : c'est l'opération de *synchronisation*, qui s'effectue encore avec le principe de l'acquittement afin de fiabiliser la transmission. On appelle dans la suite *émetteur* celui qui initie la procédure, et *récepteur* la partie adverse. On utilise la méthode dite du *three-way handshake*, dont voici le principe :

- L'émetteur envoie d'abord un message (où un bit SYN présent dans l'entête signale qu'il s'agit d'un message de synchronisation) contenant un numéro de séquence x, qui est son numéro de séquence initial.
- A réception de celui-ci, le récepteur envoie un acquittement (ACK) qui contient en plus son propre numéro de séquence initial y (SYN), ce qui indique que le prochain message qu'il s'attend à recevoir devra contenir y comme numéro d'acquittement.
- Enfin, l'émetteur acquitte ce dernier message : le numéro de séquence associé est incrémenté (soit $x + 1$).

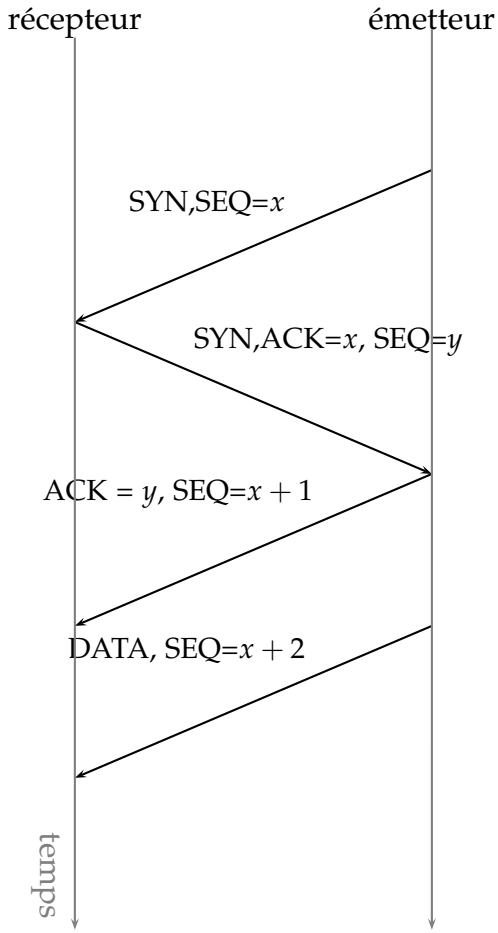


FIG. 4.3 – Le three-way handshake.

L'acquittement terminant la procédure permet au récepteur de s'assurer que la

demande de synchronisation à laquelle il vient de répondre a été acceptée, et donc de savoir quels seront les numéros de séquence utilisés dans la session qui s'amorce. En effet, au cours de l'initialisation d'une connexion, il est possible que l'un des hôtes reçoive un message de synchronisation provenant d'une ancienne tentative d'initialisation : c'est alors le dernier acquittement qui permet d'assurer la fiabilité de la transmission, les hôtes pouvant ajouter à l'en-tête des messages un bit de contrôle (RST) qui force la réinitialisation de la connexion en cas de problème. Il est en effet essentiel que le numéro initial choisi par chacun des hôtes au terme de la session soit le plus grand des numéros proposés par celui-ci, afin d'éviter toute confusion ultérieure (sinon, un paquet de donnée pourrait porter le même numéro de séquence qu'un message de synchronisation).
TCP assure ainsi la fiabilité de la transmission même si l'un des hôtes n'a plus en mémoire les numéros de séquence utilisés précédemment. Observons par exemple la situation décrite par la figure 4.4. Dans ce cas, l'émetteur envoie une demande de connexion avec le numéro de séquence 100. Celle-ci n'arrive pas à destination, mais le récepteur reçoit une demande de connexion obsolète contenant le numéro initial 90 qu'il accepte. Il renvoie alors un acquittement de cette demande, mais l'émetteur détecte à réception de cet acquittement qu'il s'agit d'une réponse à une tentative obsolète de connexion, grâce au numéro d'acquittement de celui-ci. Il décide donc d'envoyer un message de réinitialisation. Celui-ci est affublé du numéro de séquence correspondant au numéro d'acquittement du message l'ayant déclenché, puisque c'est celui qu'attend le récepteur (sinon il n'aurait pas lui-même envoyé cet acquittement, car il aurait détecté de suite qu'il s'agissait d'un message obsolète).

Le cas dual, où l'émetteur reçoit un message obsolète qu'il ne sait détecter, est corrigé grâce à la présence d'un numéro de séquence initial du côté du récepteur. La figure 4.5 en présente un exemple. Le numéro de séquence du message RST envoyé par le récepteur est dans cet exemple 291, puisque c'est le numéro de séquence que s'attend à recevoir l'émetteur : il doit donc accompagner la demande de réinitialisation afin que celle-ci soit prise en compte par l'émetteur.

Remarque : Chacun des acquittements envoyés par le client possède en fait un numéro de séquence, calculé à partir du numéro de séquence initial que celui-ci a choisi. De plus, chacun des messages envoyés par le serveur contient également un numéro d'acquittement, qui est égal au numéro de séquence du dernier message provenant du client qu'à reçu le serveur, si bien que tout message en acquitte un autre. Le numéro de séquence du client ainsi utilisé renforce la fiabilité du protocole, en permettant notamment au serveur de détecter les demandes de réinitialisation obsolètes.

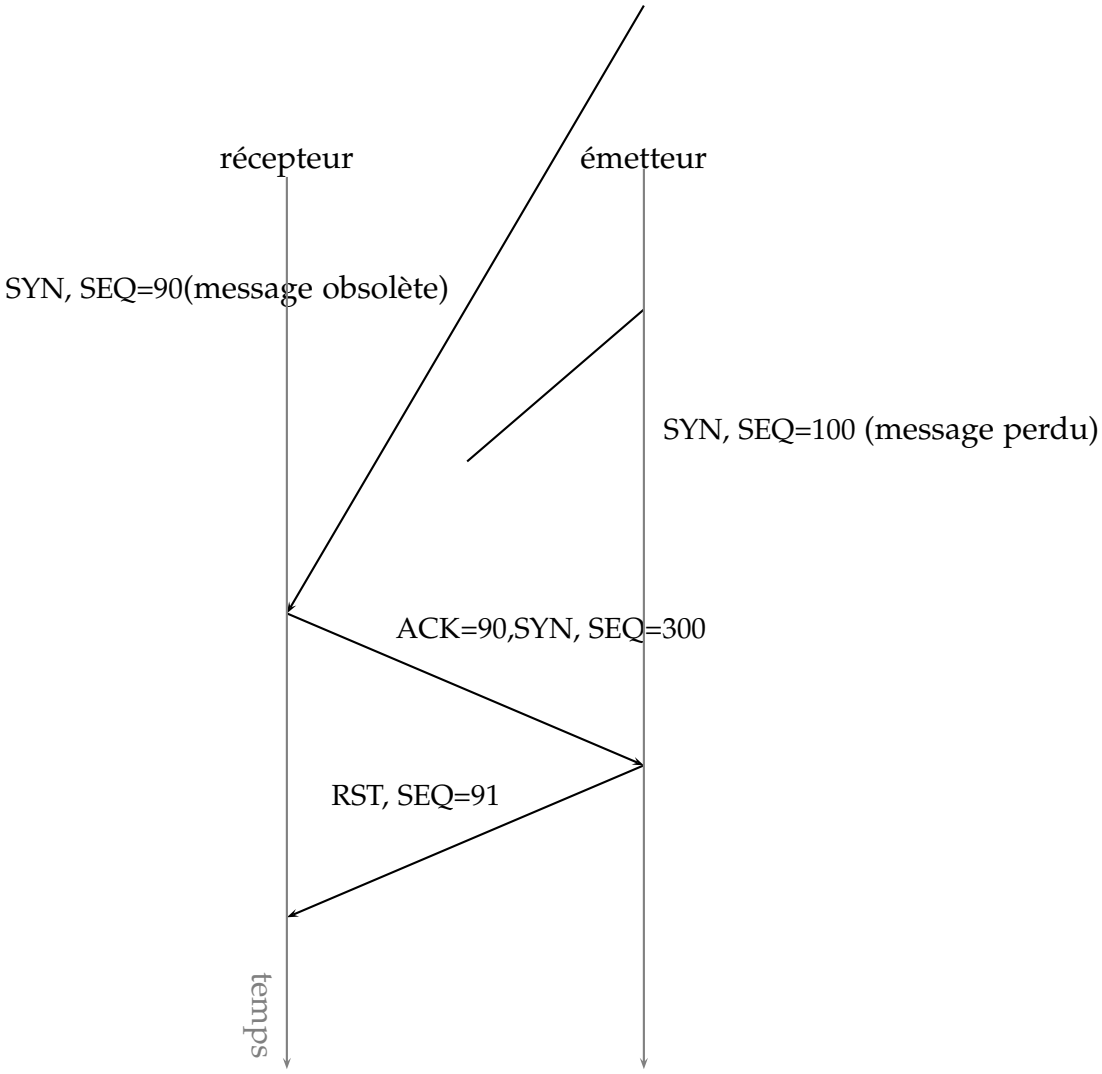


FIG. 4.4 – Une initialisation avortée par l'émetteur.

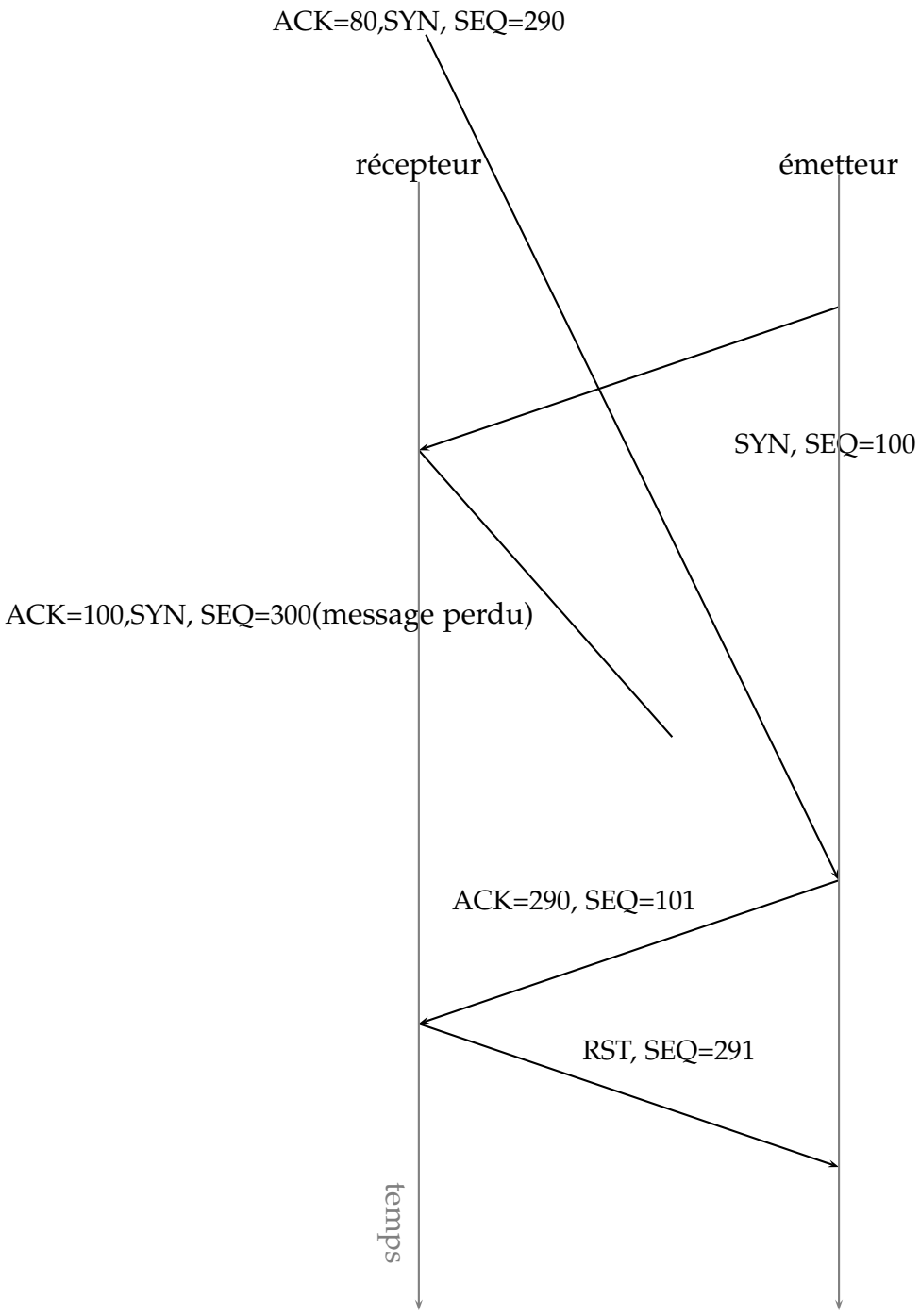


FIG. 4.5 – Une initialisation avortée par le récepteur.

4.1.3 La procédure de déconnexion

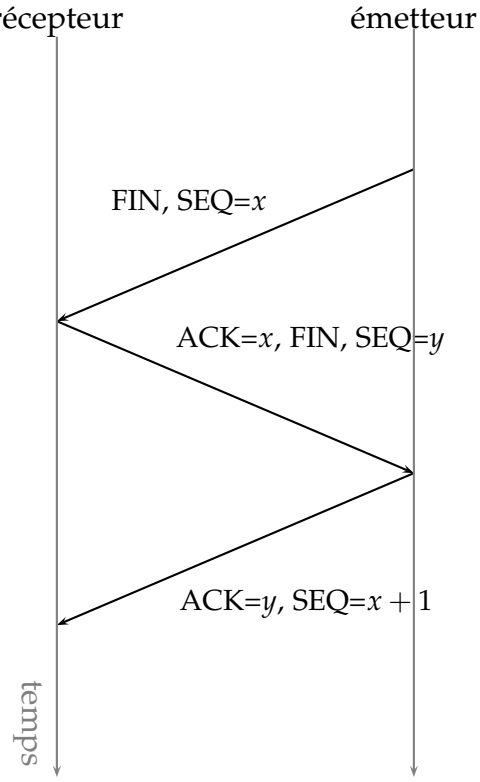


FIG. 4.6 – La procédure de déconnexion.

Elle est nécessaire, considérant que l'espace des numéros de séquence est fini, et que l'état ouvert d'une connexion empêche les numéros qu'elle utilise d'être réutilisés pour d'autres sessions. Celle-ci s'effectue en règle général de la manière suivante, avec à nouveau un échange en trois temps :

- L'émetteur envoie un message contenant un bit FIN indiquant qu'il s'agit d'un message demandant la fin de la connexion.
- Après réception, le récepteur envoie un acquittement contenant également le bit FIN.
- Enfin, l'émetteur envoie un accusé de réception du message FIN du récepteur, et il se déconnecte. Le récepteur se déconnecte lui à la réception de ce dernier acquittement.

Afin que les demandes de déconnexion provenant d'une session antérieure puissent être identifiées, ces messages sont affublés des numéros de séquence courants.

Signalons enfin qu'à l'envoi d'un message FIN, la machine déclenche une minuterie et se déconnecte lorsque le *timeout* est atteint, même si elle n'a pas reçu de confirmation de la partie adverse. Cela pose évidemment le problème des connexions semi-ouvertes, mais on ne peut de toute façon pas « acquitter des acquittements ad vitam æternam ». Le protocole spécifie en conséquence qu'un

hôte doit se déconnecter automatiquement s'il n'a pas reçu de message pendant un certain temps.

4.2 Jeux sur les formules du premier ordre

Nous allons dans cette partie associer un jeu à chaque formule du premier ordre écrite dans un ω-langage contenant un symbole de fonction pour chaque fonction récursive. Les deux joueurs, appelés $\exists$ et $\forall$, possèdent des comportements notablement différents : l'ω-validité de la formule équivaut à l'existence d'une stratégie gagnante pour $\exists$.
Ces jeux ont été introduits par Krivine, mais n'ont pas été à ce jour détaillés dans une publication (voir [17]). Ils ont été construits afin de résoudre le problème de la spécification dans le cas des formules valides (voir la section 5.2).

4.2.1 Formules ω-jouables et ω-modèles

On définit déjà l'ensemble des formules sur lesquelles on pourra définir une notion de jeu. On se donne pour cela :
- Des *variables d'individus* notées x, y...
- Des *variables entières* notées n, p...
- Un symbole de fonction f pour chaque fonction récursive $f : \mathbb{N}^k \to \mathbb{N}$ pour $k \geqslant 1$.
- Un symbole de constante n pour chaque entier n.
- Des symboles de prédicat notés R, P,... chacun possédant une arité, qui est un couple d'entiers (k, k').
- Une constante de prédicat $(0,0)$-aire notée $\bot$.

On appelle $\mathscr{L}$ le ω-langage formé par ces symboles.

Définition 4.2.1.
On appelle T l'ensemble des termes définis à partir des variables entières et des symboles de constante.
- *une variable entière est un terme.*
- *un symbole de constante est un terme.*
- *si f désigne un symbole de fonction k-aire, et si $\tau_1, \ldots, \tau_k$ désignent des termes, alors $f(\tau_1, \ldots, \tau_k)$ est un terme.*

On ne s'autorise donc pas à appliquer les symboles de fonction aux variables d'individus. Cette restriction possède une interprétation dans le cadre des porotocoles réseaux : les variables d'individus représenteront alors des numéros de séquence, sur lesquels il est hors de propos d'effectuer des opérations arithmétiques.

Définition 4.2.2.
On appelle expression égalitaire toute expression de la forme $\tau = \tau'$, où τ et τ' sont des termes.

Définition 4.2.3.
*On appelle formule ω-jouable à paramètres toute formule **close** obtenue en utilisant les règles suivantes :*

- *$\bot$ est une formule ω-jouable, dite atomique.*
- *Soient $x_1, \ldots, x_k$ des variables d'individus et $\tau_1, \ldots, \tau_{k'}$ des termes. Alors si R désigne un symbole de prédicat (k, k')-aire, $R(x_1, \ldots, x_k, \tau_1, \ldots, \tau_{k'})$ est une formule ω-jouable, dite atomique.*
- *Soient $x_1, \ldots, x_k$ des variables d'individus, $n_1, \ldots, n_{k'}$ des variables entières, $\Phi_1, \ldots, \Phi_p$ des formules ω-jouables **ou** des expressions égalitaires, et A une formule atomique. Alors $\forall x_1 \ldots \forall x_k \forall n_1 \ldots \forall n_{k'}(\Phi_1, \ldots, \Phi_p \to A)$ est une formule ω-jouable.*

Une expression égalitaire n'est pas une formule ω-jouable (en particulier elle n'est pas considérée comme une formule atomique), mais elle peut apparaître dans une formule ω-jouable « à gauche d'une flèche ».

La seule restriction sémantique induite par cette définition est que les symboles de fonctions ne s'appliquent qu'à des entiers : toute formule du premier ordre F écrite avec des symboles de $\mathscr{L}$, le symbole $=$ et possédant cette propriété est équivalente à une formule ω-jouable $\tilde{F}$. En effet, si F est de la forme $\tau = \tau'$, il suffit de poser $\tilde{F} = (\tau = \tau' \to \bot) \to \bot$. Si F est de la forme $G \to H$, et si $\tilde{H} = \forall x_1 \ldots \forall n_{k'}(\Phi_1, \ldots, \Phi_p \to A)$ de sorte que $x_1, \ldots, n_{k'}$ n'apparaissent pas dans $\tilde{G}$, on pose $\tilde{F} = \forall x_1 \ldots \forall n_{k'}(\tilde{G}, \Phi_1, \ldots, \Phi_p \to A)$ qui est équivalente à F par hypothèse d'induction.

Notations : Si $p = 0$, l'expression $\forall x_1 \ldots \forall x_k \forall n_1 \ldots \forall n_{k'}(\Phi_1, \ldots, \Phi_p \to A)$ dénotera la formule $\forall x_1 \ldots \forall x_k \forall n_1 \ldots \forall n_{k'}\, A$. Par ailleurs si $k = 0$, elle dénotera la formule $\forall n_1 \ldots \forall n_{k'}(\Phi_1, \ldots, \Phi_p \to A)$; de même si $k' = 0$. Toute formule ω-jouable est donc de la forme $\forall x_1 \ldots \forall x_k \forall n_1 \ldots \forall n_{k'}(\Phi_1, \ldots, \Phi_p \to A)$.

La définition suivante introduit une terminologie adéquate au traitement de l'égalité.

Définition 4.2.4.
Soit F une formule ω-jouable de la forme $\forall x_1 \ldots \forall x_k \forall n_1 \ldots \forall n_{k'}(\Phi_1, \ldots, \Phi_p \to A)$. On appelle :

1. *« hypothèse de F » chacune des expressions $\Phi_1, \ldots, \Phi_p$ dont la clôture est une formule ω-jouable.*
2. *« hypothèse égalitaire de F » chacune des expressions $\Phi_1, \ldots, \Phi_p$ qui est une expression égalitaire.*

Définition 4.2.5.
On appelle ω-modèle toute $\mathscr{L}$-structure $\mathscr{N}$ dans laquelle le symbole $\perp$ est interprété par le « faux », et telle que l'ensemble des éléments interprétant les symboles de constantes de $\mathscr{L}$ soit une sous-structure isomorphe au modèle standard de l'arithmétique $\mathbb{N}$. On appellera « entiers du modèle » les éléments de cette sous-structure.

On considère alors la notion usuelle de satisfaction des formules du premier ordre dans un ω-modèle : une formule $G = \forall x_1 \dots \forall x_k \forall n_1 \dots \forall n_{k'} F$ est satisfaite dans $\mathscr{N}$ (notation : $\mathscr{N} \vDash G$) si et seulement si $\mathscr{N}$ satisfait $F(\xi_1, \dots, \xi_k, p_1, \dots, p_{k'})$ quels que soient les individus $\xi_1, \dots, \xi_k$ de $\mathscr{N}$ et les entiers du modèle $p_1, \dots, p'_k$.

Définition 4.2.6.
Soit F une formule ω-jouable. Elle est dite ω-valide si et seulement si elle est satisfaite dans tous les ω-modèles.

Dans la suite on considérera des ω-modèles particuliers qui sont des *modèles de termes.* Intuitivement, le but du joueur $\forall$ sera de construire un modèle de termes qui ne satisfait pas la formule considérée.

Définition 4.2.7.
On appelle modèle de termes tout ω-modèle $\mathscr{N}$ dont l'ensemble de base est formé par les variables d'individus considérées et les symboles de constante de $\mathscr{L}$.

Définition 4.2.8.
Soit F une formule ω-jouable de la forme $\forall x_1 \dots \forall x_k \forall n_1 \dots \forall n_{k'} (\Phi_1, \dots, \Phi_p \to A)$. On appelle « uplet compatible avec F » tout $(k+k')$-uplet $(\xi_1, \dots, \xi_{k+k'})$ où $\xi_1, \dots, \xi_k$ sont des variables d'individu et $\xi_{k+1}, \dots, \xi_{k+k'}$ sont des symboles de constante.

Une formule $\forall x_1 \dots \forall x_k \forall n_1 \dots \forall n_{k'} F$ est donc satisfaite dans un modèle de termes $\mathscr{N}$ si et seulement si $\mathscr{N} \vDash F(\xi_1, \dots, \xi_{k+k'})$ quel que soit le uplet compatible $\xi_1, \dots, \xi_{k+k'}$.

4.2.2 Le jeu

Nous allons associer à chaque formule ω-jouable un jeu défini entre deux joueurs : le premier sera noté $\forall$ (ou Abélard) et la seconde $\exists$ (ou Héloïse). Intuitivement, la joueuse $\exists$ cherche à défendre la formule considérée en essayant d'en construire une preuve, alors que le joueur $\forall$ l'attaque en essayant de construire un modèle de termes dans lequel celle-ci n'est pas satisfaite.

Le jeu se fait autour de trois ensembles de formules ω-jouables notés $\mathscr{U}$, $\mathscr{V}$ et $\mathscr{A}$, qui évoluent au cours d'une partie. L'ensemble $\mathscr{A}$ ne peut contenir que des formules atomiques. Les ensembles $\mathscr{U}$ et $\mathscr{A}$ sont croissants (au sens de l'inclusion) au cours d'une partie. $\mathscr{U}$ est l'ensemble des choix possibles de $\exists$ au moment considéré, et $\mathscr{V}$ l'ensemble des choix possibles de $\forall$.

Dans le jeu associé à une formule F, la partie débute avec $\mathscr{U} = \{F \to \bot\}$, $\mathscr{V} = \{F\}$ et $\mathscr{A} = \{\bot\}$. C'est le joueur $\forall$ qui commence la partie.

On se place maintenant à un moment quelconque d'une partie où c'est au tour de $\forall$ de jouer.
Il choisit une formule $\Phi \in \mathscr{V}$, $\Phi = \forall x_1 \dots \forall x_k \forall n_1 \dots \forall n_{k'}(\Psi_1, \dots, \Psi_p \to A)$, ainsi qu'un uplet $(\xi_1, \dots, \xi_m)$ compatible.
Si l'une des hypothèses égalitaires de Φ n'est pas satisfaite dans $\mathbb{N}$ par le uplet $(\xi_1, \dots, \xi_m)$, le jeu s'arrête et $\forall$ a perdu la partie.
Sinon, la formule $A(\xi_1, \dots, \xi_m)$ est ajoutée à l'ensemble $\mathscr{A}$, et les hypothèses de Φ sont instanciées en $(\xi_1, \dots, \xi_m)$ puis ajoutées à l'ensemble $\mathscr{U}$.
C'est ensuite à la joueuse $\exists$ de jouer.

Elle choisit une formule $\Psi \in \mathscr{U}$, $\Psi = \forall x_1 \dots \forall x_k \forall n_1 \dots \forall n_{k'}(\Phi_1, \dots, \Phi_p \to A)$ ainsi qu'un uplet $(\xi_1, \dots, \xi_m)$ compatible tel que :
- la formule atomique $A(\xi_1, \dots, \xi_m)$ est dans l'ensemble $\mathscr{A}$;
- chacune des hypothèses égalitaires de Ψ est satisfaite dans $\mathbb{N}$ par le uplet $(\xi_1, \dots, \xi_m)$.

L'ensemble $\mathscr{V}$ devient alors l'ensemble des hypothèses de Ψ instanciées en $(\xi_1, \dots, \xi_m)$.
C'est ensuite au tour de $\forall$ de jouer.

Il faut noter ici que la joueuse $\exists$ peut toujours satisfaire les contraintes qui lui sont imposées, puisqu'elle peut toujours choisir la formule $F \to \bot$.

La joueuse $\exists$ l'emporte si et seulement si la partie s'arrête au bout d'un temps fini. C'est notamment le cas si $\forall$ ne peut plus jouer à un instant donné :
- soit parce qu'à cet instant l'ensemble $\mathscr{V}$ est vide, ce qui signifie que $\exists$ a pu choisir dans $\mathscr{U}$ une formule atomique qui était déjà dans $\mathscr{A}$;
- soit parce que chaque formule dans $\mathscr{V}$ possède des hypothèses égalitaires qui ne peuvent être simultanément satisfaites dans $\mathbb{N}$.

Le joueur $\forall$ l'emporte dans le cas contraire, c'est-à-dire lorsque la partie dure indéfiniment.

On peut démontrer que ces jeux sont déterminés, c'est-à-dire que pour chacun d'entre eux, l'un des joueurs possède une stratégie gagnante, ce qui résulte du théorème suivant.

Théorème 4.2.9.
Soit F une formule ω-jouable.
i) Tout modèle de termes $\mathscr{N}$ ne satisfaisant pas F donne une stratégie gagnante pour le joueur $\forall$.
ii) Il existe une stratégie pour la joueuse $\exists$ ayant la propriété suivante : chaque partie perdue par $\exists$ en employant cette stratégie donne un modèle de termes $\mathscr{N}$ qui ne satisfait pas F.

Démonstration.
i) Soit une F une formule ω-jouable et $\mathscr{N}$ un modèle de termes tel que $\mathscr{N} \not\models F$. Nous allons donner une stratégie pour $\forall$ possédant la propriété suivante :

A chaque étape d'une partie où $\forall$ applique celle-ci,
- il existe une formule Φ dans $\mathscr{V}$ telle que $\mathscr{N} \not\models \Phi$;
- pour toute formule A dans $\mathscr{A}$, on a $\mathscr{N} \not\models A$;
- pour toute formule Ψ dans $\mathscr{U}$, on a $\mathscr{N} \models \Psi$.

Nous allons construire récursivement une telle stratégie.
La propriété énoncée est évidemment vraie au début de la partie, puisqu'alors $\mathscr{U} = \{F \to \bot\}$, $\mathscr{V} = \{F\}$ et $\mathscr{A} = \{\bot\}$.
On se place maintenant à un moment donné d'une partie où c'est au tour de $\forall$ de jouer. On suppose donc qu'à ce stade il existe dans $\mathscr{V}$ une formule Φ telle que $\mathscr{N} \models \neg\Phi$. Donc si $\Phi = \forall x_1 \dots \forall x_k \forall n_1 \dots \forall n_{k'}(\Psi_1, \dots, \Psi_p \to A)$, il existe un uplet $(\xi_1, \dots, \xi_m)$ compatible tel que $\mathscr{N}$ satisfait les expressions suivantes : $\Psi_1(\xi_1, \dots, \xi_m), \dots, \Psi_p(\xi_1, \dots, \xi_m)$ et $\neg A(\xi_1, \dots, \xi_m)$. On impose au joueur $\forall$ de choisir un tel couple $(\Phi, (\xi_1, \dots \xi_m))$, si bien qu'il ne perd pas la partie à cet instant, et que les nouveaux ensembles $\mathscr{U}$, $\mathscr{V}$ et $\mathscr{A}$ alors obtenus conservent la propriété considérée.
Considérons maintenant un moment donné où c'est au tour de $\exists$ de jouer, et où la propriété énoncée est satisfaite. Elle choisit une formule $\Psi \in \mathscr{U}$, $\Psi = \forall x_1 \dots \forall x_k \forall n_1 \dots \forall n_{k'}(\Phi_1, \dots, \Phi_p \to A)$ ainsi qu'un uplet $(\xi_1, \dots, \xi_m)$ compatible. Cela signifie que $A(\xi_1, \dots, \xi_m)$ était déjà dans $\mathscr{A}$, et donc la formule atomique $A(\xi_1, \dots, \xi_m)$ n'est pas satisfaite par $\mathscr{N}$. L'hypothèse de récurrence assurant que par ailleurs $\mathscr{N} \models \Psi$, on en déduit que l'une au moins des expressions $\Phi_1(\xi_1, \dots, \xi_m), \dots, \Phi_p(\xi_1, \dots, \xi_m)$ n'est pas satisfaite par $\mathscr{N}$, et cette expression est fatalement une formule ω-jouable puisque $\exists$ a pu choisir $(\Psi, (\xi_1, \dots, \xi_m))$. La propriété considérée est donc encore vérifiée par les nouveaux ensembles obtenus, ce qui termine la construction de la stratégie.

Pour conclure, il suffit de remarquer que cette stratégie est gagnante pour $\forall$, puisqu'elle lui permet de toujours choisir une formule dans $\mathscr{V}$ et un uplet d'éléments de $\mathscr{N}$ qui en satisfait les hypothèses égalitaires : la partie dure donc indéfiniment.

ii) Fixons une énumération des couples de la forme $(\Psi, (\xi_1, \dots, \xi_m))$, où $\Psi = \forall x_1 \dots \forall x_k \forall n_1 \dots \forall n_{k'}(\Phi_1, \dots, \Phi_p \to A)$ est une formule ω-jouable et $(\xi_1, \dots, \xi_m)$ un uplet compatible.

On considère la stratégie suivante pour la joueuse : *à chaque étape, $\exists$ choisit parmi toutes les paires $(\Psi, (\xi_1, \dots, \xi_m))$ autorisées (par les règles du jeu) la première parmi celles qu'elle n'a pas encore choisies dans cette partie ; s'il n'existe aucune paire de la sorte, $\exists$ déclare forfait pour la partie en cours.*

Considérant une partie, qualifions d'*acceptable* toute paire $(\Psi, (\xi_1, \ldots, \xi_m))$ qui est telle que :
- Ψ est une formule ω-jouable de la forme $\forall x_1 \ldots \forall n_{k'}(\Phi_1, \ldots, \Phi_p \rightarrow A)$;
- $(\xi_1, \ldots, \xi_m)$ est un uplet compatible avec Ψ ;
- chacune des hypothèses égalitaires de Ψ est satisfaite par $(\xi_1, \ldots, \xi_m)$;
- à un moment donné dans cette partie, Ψ est dans $\mathscr{U}$ et $A(\xi_1, \ldots, \xi_m)$ est dans $\mathscr{A}$.

On peut démontrer que dans une partie remportée par $\forall$ où $\exists$ applique la stratégie considérée, toute paire acceptable est choisie par la joueuse à une certaine étape. En effet, si $(\Psi, (\xi_1, \ldots, \xi_m))$ est le premier contre-exemple, on considère le premier moment de la partie où les formules Ψ et $A(\xi_1, \ldots, \xi_m)$ sont respectivement dans $\mathscr{U}$ et $\mathscr{A}$, et où toutes les paires inférieures ont déja été choisies. La stratégie considérée impose alors à $\exists$ de choisir cette paire : on a une contradiction.

Considérons maintenant une partie qui est perdue par $\exists$ alors qu'elle applique la stratégie considérée. On appelle $\mathscr{N}$ le modèle de termes obtenu en interprétant chaque symbole de prédicat R, d'arité (k, k'), de la manière suivante :

$(\xi_1, \ldots, \xi_{k+k'}) \in R$ ssi $R(\xi_1, \ldots, \xi_{k+k'})$ n'entre pas dans $\mathscr{A}$ au cours de la partie.

Nous allons prouver par induction sur l'ensemble des formules ω-jouables à paramètres que :
- si Ψ entre dans $\mathscr{U}$ au cours de la partie, alors $\mathscr{N} \vDash \Psi$;
- si Φ est à un moment donné choisie par $\forall$, alors $\mathscr{N} \nvDash \Phi$.

Il en résultera que $\mathscr{N}$ ne satisfait pas F, puisque la formule F est nécessairement choisie par le joueur $\forall$ au premier tour, ce qui terminera la démonstration.

Soit Ψ une formule qui entre dans $\mathscr{U}$ au cours de la partie.
Supposons que Ψ est atomique. Si Ψ n'était pas satisfaite par $\mathscr{N}$ cela signifierait que Ψ entre dans $\mathscr{A}$ à un moment donné, et donc que la paire $(\Psi, \emptyset)$ est acceptable. Mais alors la joueuse la choisirait à un moment donné, et remporterait aussitôt la partie, ce qui est une contradiction.
Supposons maintenant que $\Psi = \forall x_1 \ldots \forall x_k \forall n_1 \ldots \forall n_{k'}(\Phi_1, \ldots, \Phi_p \rightarrow A)$. Il nous faut montrer que pour tout uplet $(\xi_1, \ldots, \xi_m)$ compatible, $\mathscr{N}$ satisfait la formule $\Phi_1(\xi_1, \ldots, \xi_m), \ldots, \Phi_p(\xi_1, \ldots, \xi_m) \rightarrow A(\xi_1, \ldots, \xi_m)$. C'est évident si l'une des hypothèses égalitaires de cette formule n'est pas satisfaite par le uplet choisi, ou si $A(\xi_1, \ldots, \xi_m)$ n'entre jamais dans $\mathscr{A}$. Sinon, la paire $(\Psi, (\xi_1, \ldots, \xi_m))$ est acceptable et est donc choisie par $\exists$ à un moment donné. A cet instant, $\mathscr{V}$ contiendra celles parmi les expressions $\Phi_1(\xi_1, \ldots, \xi_m), \ldots, \Phi_p(\xi_1, \ldots, \xi_m)$ qui sont des formules ω-jouables. $\forall$ choisira alors l'une de ces formules (il y en a au moins une, puisque $\forall$ est supposé remporter la partie) et l'hypothèse d'in-

duction assure que celle-ci n'est pas satisfaite par $\mathscr{N}$, ce qui termine la démonstration dans ce cas.

Soit Φ une formule qui est choisie par $\forall$ à un moment donné de la partie.
Si Φ est atomique, le résultat est trivial puisque Φ entre dans $\mathscr{A}$ à cet instant. Sinon, on considère qu'elle s'écrit $\forall x_1 \dots \forall n_{k'}(\Psi_1, \dots, \Psi_p \to A)$, que le joueur $\forall$ choisit $(\xi_1, \dots, \xi_m)$, et ajoute donc $A(\xi_1, \dots, \xi_m)$ à $\mathscr{A}$ et les hypothèses (instanciées) de Φ à $\mathscr{U}$. Par hypothèse d'induction il vient que $\mathscr{N}$ satisfait chaque hypothèse (instanciée) de Φ. Par ailleurs, les hypothèses égalitaires de Φ sont également satisfaites par ce uplet, puisque $\forall$ est supposé remporter la partie. Or on a par définition que $\mathscr{N} \not\models A(\xi_1, \dots, \xi_m)$ puisque $A(\xi_1, \dots, \xi_m)$ entre dans $\mathscr{A}$, et donc $\mathscr{N} \models \neg\Phi$, ce qui termine la preuve.

□

Dans le cas d'une formule ω-valide, on dispose d'un résultat plus fort qui possède une interprétation dans le cadre des protocoles réseaux (voir la section 4.3).

Théorème 4.2.10.
Soit F une formule ω-jouable, et ξ_0 une variable d'individus.
F est ω-valide si, et seulement si, la joueuse $\exists$ possède une stratégie gagnante ayant la propriété suivante :
$\exists$ ne choisit à chaque instant que des individus appartenant à la sous-structure engendrée par ξ_0 et par les individus choisis plus tôt dans la partie par $\forall$.

Démonstration. La condition suffisante résulte du point *i*) du théorème 4.2.9.
Pour démontrer la condition nécessaire, on fixe d'abord une énumération des couples de la forme $(\Psi, (\xi_1, \dots, \xi_m))$. On considère ensuite la stratégie suivante pour la joueuse :
à chaque étape, parmi toutes les paires $(\Psi, (\xi_1, \dots, \xi_m))$ autorisées telles que $\xi_1, \dots, \xi_m$ appartiennent à la sous-structure engendrée par ξ_0 et par les individus choisis auparavant par $\forall$, $\exists$ choisit la première parmi celles qu'elle n'a pas encore choisies dans cette partie ; s'il n'existe aucune paire de la sorte, $\exists$ déclare forfait pour la partie en cours.
Il suffit alors d'ajouter dans la définition des paires acceptables la contrainte suivante : *$\xi_1, \dots, \xi_m$ appartiennent à la sous-structure engendrée par ξ_0 et par les individus choisis auparavant par $\forall$.*
On démontre alors en raisonnant comme au point *ii*) du théorème 4.2.9 que, s'il existe une partie perdue par $\exists$ en employant cette stratégie, alors la sous-structure engendrée par les individus choisis par $\forall$ au cours de la partie ne satisfait pas F. Cela démontre le résultat, et assure en particulier que chaque paire acceptable $(\Psi, (\xi_1, \dots, \xi_m))$ est choisie au plus une fois au cours d'une partie. □

4.2.3 Un premier exemple

On représentera une partie dans le jeu associé à une formule F donnée par un tableau, qui indique à chaque étape le contenu des ensembles $\mathscr{U}$, $\mathscr{V}$ et $\mathscr{A}$ ainsi que les choix effectués à ce moment-là par celui des deux joueurs dont c'est le tour. On ne réécrira néanmoins pas à chaque ligne le contenu des ensembles $\mathscr{U}$ et $\mathscr{A}$: ceux-ci étant croissants, on se contente d'écrire quelles sont les formules qui ont été rajoutées à ces ensembles au cours du dernier tour de jeu, séparées par un point-virgule.

On peut donner un premier exemple, en considérant la formule propositionnelle exprimant le *modus ponens* : $A, (A \to B) \to B$.
On représente dans le tableau ci-dessous la stratégie gagnante de la joueuse $\exists$ qui donne lieu à la partie la plus courte possible :

$\mathscr{U}$	$\mathscr{V}$	$\mathscr{A}$	Action
$\neg(A, (A \to B) \to B)$	$A, (A \to B) \to B$	$\bot$	$\forall$ choisit $A, (A \to B) \to B$
$A\,;\, A \to B$	inchangé	B	$\exists$ choisit $A \to B$
inchangé	A	inchangé	$\forall$ choisit A
inchangé	inchangé	A	$\exists$ choisit A
inchangé	vide	inchangé	$\forall$ ne peut rien choisir et perd la partie

Il existe évidemment une grande variété de parties possibles. $\exists$ l'emporte dès qu'elle choisit la formule A, et pour ce faire elle doit au préalable avoir choisi la formule $A \to B$. Mais les règles du jeu lui permettent de choisir les formules $A \to B$ et $\neg(A, (A \to B) \to B)$ autant de fois qu'elle le désire.

4.3 Application à la spécification de protocoles réseaux

Les principes de base de la modélisation sont les suivants :

1. Le jeu associé à une formule ω-valide donnée peut être interprété comme spécifiant un protocole réseau. Les règles du jeu décrivent ce que le protocole autorise (ou impose) au client et au serveur. Une partie associée à un tel jeu correspond à une session particulière.
2. Le joueur $\forall$ modélise l'expéditeur des messages considérés, alors que la joueuse $\exists$ modélise l'ensemble formé par le destinataire (de ces messages) et le canal par lequel ceux-ci transitent. C'est-à-dire que dans une partie donnée, le comportement du joueur $\forall$ décrit le comportement de l'expéditeur au cours d'une session donnée, alors que le comportement de la joueuse $\exists$ décrit le comportement du destinataire ainsi que les différents phénomènes ayant lieu sur le canal au cours de cette session.
   On considère donc que **les protocoles sont décrits du point de vue de l'expéditeur**, celui-ci ne pouvant distinguer le destinataire du canal les reliant.
3. Les formules atomiques $Rx_1, \ldots, x_n$ symbolisent des messages : le prédicat R représente un paquet de données, alors que les variables $x_1, \ldots, x_n$ représentent les champs de son en-tête. Ceux parmi les x_i qui sont des variables d'individus représentent des numéros de séquence, les autres représentant des quantités entières (par exemple le numéro d'ordre du paquet) sur lesquelles on sera amené à effectuer des opérations arithmétiques simples.
4. Lorsque l'un des joueurs doit choisir une formule dans l'ensemble qui lui est associé ($\mathscr{V}$ ou $\mathscr{U}$), on ne suppose a priori pas qu'il connaît les paramètres apparaissant dans ces formules qui ont été choisis par l'autre joueur. On suppose seulement qu'il connaît la structure de ces formules, ainsi que leur atome terminal.
5. L'ensemble $\mathscr{U}$ représente à un instant donné l'ensemble des messages qui ont été envoyés depuis le début de la session. Sa croissance modélise le fait que tout message envoyé à un moment donné est ensuite susceptible d'arriver à destination à n'importe quel instant.

4.3.1 Envoi d'un paquet avec acquittement

Cet exemple est particulièrement important, car il modélise les fondements de tout protocole réseau, et donne un aperçu de ce que ces jeux permettent de modéliser. Dans cette section, la joueuse $\exists$ représentera l'ensemble formé par le client et le canal, alors que le joueur $\forall$ représentera le serveur (ce sera toujours le cas, excepté dans le jeu décrivant la procédure d'initialisation où les rôles seront inversés).
On considère la formule valide $F = \forall x(\forall y Py \rightarrow Px)$; la joueuse $\exists$ possède donc une stratégie gagnante pour les jeux qui lui sont associés. Le tableau suivant désigne l'ensemble des parties gagnantes pour $\exists$ possibles, en fonction de deux entiers n et i.

$\mathscr{U}$	$\mathscr{V}$	$\mathscr{A}$	Action
$F \rightarrow \bot$	F	$\bot$	$\forall$ choisit F et x_1
$\forall y Py$	F	Px_1	$\exists$ choisit $F \rightarrow \bot$
inchangé	F	inchangé	$\forall$ choisit F et x_2
inchangé	F	Px_2	$\exists$ choisit $F \rightarrow \bot$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
inchangé	F	inchangé	$\forall$ choisit F et x_n
inchangé	F	Px_n	$\exists$ choisit $\forall y Py$ et $y = x_i$
inchangé	vide	inchangé	$\forall$ ne peut rien choisir et perd la partie

Le joueur $\forall$ est à chaque instant obligé de choisir la formule F ainsi qu'un individu y. Tant que la joueuse $\exists$ choisit la formule $F \rightarrow \bot$, elle force $\forall$ à refaire un tel choix. Si à un moment donné elle choisit la formule $\forall y Py$ et un individu x_i tel que $Px_i \in \mathscr{A}$, elle remporte aussitôt la partie.

On considère que ce tableau modélise une session de communication, où le serveur $\forall$ veut envoyer un message contenant les données P au client $\exists$.
Il expédie d'abord ce message avec l'en-tête x_1. Si la joueuse choisit alors la formule $F \rightarrow \bot$, on considère que le serveur n'a reçu aucun acquittement avant la fin du timeout, et décide de renvoyer les données avec un en-tête x_2. Le jeu se poursuit jusqu'à ce que $\exists$ choisisse $\forall y Py$ et un individu x_i choisi précédemment par $\forall$. On considère dans ce cas que le client avait déjà reçu le paquet de données contenant l'en-tête x_i, qu'il avait envoyé l'acquittement correspondant, et qu'à cet instant celui-ci est reçu par le serveur. Ce dernier considère alors que

le message est arrivé à destination et se déconnecte : la session est terminée.

On peut donc considérer que cette formule décrit un protocole imposant au serveur de renvoyer les données tant qu'il n'a pas reçu d'acquittement, le client étant libre d'envoyer ou pas des acquittements, mais il doit le faire s'il souhaite remporter la partie.
Mais au contraire, on peut aussi considérer que lorsque $\exists$ choisit la formule $F \to \bot$ c'est le client qui redemande l'envoi des données, et que l'on a donc affaire à un serveur qui ne se préoccupe pas des acquittements que celui-ci envoie. On peut alors interpréter ces règles comme décrivant un protocole où le client doit demander les données tant qu'il ne les a pas reçues, le serveur se devant simplement de répondre aux requêtes qu'il reçoit.

Cette description ne détaille pas la cause de l'échec de l'acquittement d'un message donné. On ne sait pas si ce message n'est pas arrivé jusqu'au client, si celui-ci l'a reçu et qu'il n'a pas envoyé d'acquittement, ou si l'acquittement n'est pas arrivé jusqu'au serveur. Mais en tout les cas ce phénomène est dû au comportement du client ou à celui de la ligne, et est donc modélisé par la joueuse. Une partie décrit donc une session de communication du point de vue du serveur, qui ne peut jamais dissocier ces deux acteurs.

Remarque : On a coutume de considérer qu'il est impossible d'énumérer tous les acquittements possibles, ce qui permet d'assurer la fiabilité du principe de l'acquittement face à certaines attaques. On peut donc interpréter les individus non-entiers de l'ω-modèle considéré comme formant l'ensemble des numéros de séquence possibles, ceux-ci n'étant a priori pas dénombrables. Par ailleurs, on peut considérer que dans une stratégie raisonnable pour la joueuse $\exists$, celle-ci n'est pas amenée à choisir n'importe quel individu : soit elle reçoit le message et renvoie l'individu qu'elle a reçu, soit elle ne le reçoit pas auquel cas elle choisit un individu ξ_0 signifiant ce fait. On peut donc souhaiter imposer que les individus choisis par la joueuse $\exists$ à un moment donné appartiennent à la sous $\mathscr{L}$-structure engendrée par ξ_0 et par les individus déjà joués par $\forall$. Le théorème 4.2.10 assure l'existence d'une stratégie gagnante pour la joueuse sous cette contrainte, puisque la formule considérée ici est ω-valide.

4.3.2 Bonne fondation et envoi de multiples paquets

On considère la formule suivante qui exprime la bonne fondation du successeur dans les ω-modèles :

$$\forall X \forall n \left(\forall p \left(\forall m \{ sm = p \to Xm \} \to Xp \right) \to Xn \right).$$

On considère dans cette section le cas particulier suivant, où Xn est une formule de la forme $\forall x Pn, x$. La formule obtenue est alors ω-valide :

$$\forall n \forall x \left(\forall p \forall x' \left(\forall m \forall y \{ sm = p \to Pm, y \} \to Pp, x' \right) \to Pn, x \right).$$

Le serveur a en général besoin de segmenter les données qu'il envoie au client en de multiples paquets, la taille d'un paquet étant spécifiée par le protocole de communication employé. Cette formule décrit un protocole permettant au serveur de choisir un entier n, puis d'envoyer $n+1$ messages, en attendant à chaque fois d'avoir reçu l'acquittement associé au k-ième avant d'envoyer le $(k+1)$-ième.
Le prédicat P représente alors l'ensemble des données que souhaite envoyer le serveur, Pn représentant le n-ième paquet de données. Ces numéros permettent au client de recomposer *in fine* les données envoyées par le serveur. Dans le protocole décrit ici, le paquet de numéro $n-k$ est le $(k+1)$-ième à être envoyé.
Le tableau ci-dessous représente la partie la plus simple possible dans le cas où $\forall$ choisit $n = 0$.
On pose $G(x', p) = \forall m \forall y \{ sm = p \to Pm, y \} \to Pp, x'$.

$\mathscr{U}$	$\mathscr{V}$	$\mathscr{A}$	Action
$F \to \bot$	F	$\bot$	$\forall$ choisit F, $n = 0$ et x_1
$\forall x' \forall p G(x', p)$	inchangé	P_0, x_1	$\exists$ choisit $\forall x' \forall p G(x', p)$ $x' = x_1$ et $p = 0$
inchangé	$\forall m \forall y \{ sm = 0 \to Pm, y \}$	inchangé	$\forall$ ne peut choisir m tel que $sm = 0$, il perd donc la partie.

Ce tableau s'interprète comme le précédent, et décrit un cas où le serveur décide d'envoyer un seul message de données, celui-ci étant acquitté de suite par le client. On peut encore modéliser tous les phénomènes pouvant se produire lors de l'envoi d'un message avec acquittement, puisque $\exists$ peut toujours choisir la formule $F \to \bot$ et forcer ainsi $\forall$ à renvoyer le message.

Considérons maintenant la partie la plus simple possible où $\forall$ choisit $n = 1$, c'est-à-dire un cas où le serveur décide d'envoyer deux paquets de données. Celle-ci est représentée par le tableau suivant.

$\mathscr{U}$	$\mathscr{V}$	$\mathscr{A}$	Action
$F \to \bot$	F	$\bot$	$\forall$ choisit F, $n = 1$ et x_1
$\forall x' \forall p G(x', p)$	inchangé	P_1, x_1	$\exists$ choisit $\forall x' \forall p G(x', p)$, $x' = x_1$ et $p = 1$
inchangé	$\forall m \forall y \{sm = 1 \to Pm, y\}$	inchangé	$\forall$ choisit $\forall m \forall y \{sm = 1 \to Pm, y\}$, $m = 0$ et y_1
inchangé	inchangé	P_0, y_1	$\exists$ choisit $\forall x' \forall p G(x', p)$ $x' = y_1$ et $p = 0$
inchangé	$\forall m \forall y \{sm = 0 \to Pm, y\}$	inchangé	$\forall$ ne peut satisfaire la condition $sm = 0$ et perd la partie.

Cette partie modélise une situation où le serveur reçoit chaque acquittement dès le premier envoi des données.
La seule différence avec le jeu considéré précédemment est que $\forall$ peut interrompre la partie lorsqu'il choisit la formule $\forall m \forall y \{sm = 1 \to Pm, y\}$, en choisissant un entier $m \neq 0$. Cela s'interprète comme le fait que le serveur peut à chaque moment refuser d'envoyer un paquet et interrompre la connexion en cours : le protocole considéré autorise donc le déni de service.
Il existe encore une fois une grande variété de parties possibles, chaque message étant renvoyé tant qu'aucun acquittement n'est reçu par le serveur. A première vue, il s'agit du jeu précédent mais joué deux fois bout-à-bout : une fois que $\exists$ a remporté une partie dans ce jeu, celui-ci recommence et $\exists$ doit encore remporter cette deuxième partie. Il n'en est rien, puisque la joueuse $\exists$ peut choisir la formule $F \to \bot$ à n'importe quel moment de la partie, même lorsqu'elle a déjà reçu et acquitté un certain nombre de messages : on interprète cette action comme modélisant le fait que le destinataire a envoyé une demande de réinitialisation des envois à l'expéditeur, celui-ci étant obligé de l'accepter.
Plus généralement, les règles du jeu permettent toujours à $\exists$ de choisir une formule « déjà jouée précédemment ». Cela permet de modéliser l'arrivée du doublon d'un acquittement, après que les données concernées aient été acquittés. Illustrons ce phénomène dans le cas où le serveur décide d'envoyer trois paquets de données.

$\mathscr{U}$	$\mathscr{V}$	$\mathscr{A}$	Action
$F \to \bot$	F	$\bot$	$\forall$ choisit F, $n = 2$ et x_1 : **envoi du premier paquet.**
$\forall x' \forall p G(x', p)$	inchangé	P_2, x_1	$\exists$ choisit $F \to \bot$: **le serveur n'a pas reçu d'acquittement.**
inchangé	F	inchangé	$\forall$ choisit F, $n = 2$ et x_2 : **second envoi du premier paquet.**
inchangé	inchangé	P_2, x_2	$\exists$ choisit $\forall x' \forall p G(x', p)$, $x' = x_2$ et $p = 2$: **acquittement du premier paquet.**
inchangé	$\forall m \forall y \{sm = 2 \to Pm, y\}$	inchangé	$\forall$ choisit $m = 1$ et y_1 : **envoi du deuxième paquet.**
inchangé	inchangé	P_1, y_1	$\exists$ choisit $\forall x' \forall p G(x', p)$, $x' = y_1$ et $p = 1$: **acquittement du deuxième paquet.**
inchangé	$\forall m \forall y \{sm = 1 \to Pm, y\}$	inchangé	$\forall$ choisit $m = 0$ et z_1 : **envoi du troisième paquet.**
inchangé	inchangé	P_0, z_1	$\exists$ choisit $\forall x' \forall p G(x', p)$, $x' = x_1$ et $p = 2$: **arrivée du premier acquittement du premier paquet.**
inchangé	$\forall m \forall y \{sm = 2 \to Pm, y\}$	inchangé	$\forall$ choisit $m = 1$, $y = y_2$ et $\forall m \forall y \{sm = 2 \to Pm, y\}$: **second envoi du deuxième paquet.**
$\vdots$	$\vdots$	$\vdots$	$\vdots$

A l'avant dernière ligne de ce tableau, on interprète le choix de $\exists$ comme modélisant la réception par le serveur du doublon d'un acquittement qu'il avait déjà reçu auparavant : les règles du jeu lui imposent alors de renvoyer les données qu'appellent cet acquittement.

Dans le protocole TCP, chaque acquittement contient le numéro de séquence du prochain paquet attendu par le client. Si le paquet appelé par un acquittement avait déjà été envoyé et acquitté, le protocole TCP ne spécifie pas que le serveur doit le renvoyer. C'est là une différence avec les protocoles que nous décrivons : le serveur ne possède ici pas de mémoire, puisque l'ensemble $\mathscr{V}$

est totalement réactualisé à chaque tour de jeu. Le joueur $\forall$ ne peut donc jouer qu'en fonction du dernier message qu'il a reçu. La conséquence est qu'il doit à réception de chaque acquittement envoyer le paquet demandé par celui-ci, quand bien même celui-ci a déjà été envoyé et acquitté. Cela correspond au serveur le plus simple possible, qui ne possède pas de mémoire et se comporte de manière déterministe en fonction du dernier message qu'il a reçu, excepté que le déni de service lui est autorisé et est laissé à son entière appréciation. Un tel serveur peut sembler simpliste, mais il a l'avantage d'être sûr.
Au contraire, la joueuse $\exists$ possède la mémoire de tout ce qui s'est passé au cours de la partie, puisque l'ensemble $\mathscr{U}$ est croissant. Cela lui permet de modéliser les phénomènes se passant sur le réseau, où tout message envoyé reste susceptible d'arriver à destination à n'importe quel moment. Mais cela a également pour conséquence que le client n'est pas obligé d'acquitter un message qu'il avait déjà acquitté précédemment, si cet acquittement a été reçu par le serveur (ce qu'il sait si le serveur a depuis envoyé un autre paquet). La partie peut donc se terminer de la manière suivante : le client reçoit à nouveau le premier paquet, mais conscient qu'il s'agit d'un doublon, il décide de renvoyer une demande du troisième paquet (qui n'est autre que l'acquittement associé au second paquet, acquittement qu'il a en sa possession).
Aux contraintes imposées à $\forall$ correspondent des libertés octroyées à $\exists$; c'est là une seconde différence avec le protocole TCP, qui spécifie que lorsque le client reçoit un message dont le numéro de séquence ne se trouve pas dans la fenêtre courante (par exemple un message qu'il a déjà acquitté), celui-ci doit malgré tout renvoyer un acquittement contenant le numéro de séquence du prochain paquet attendu.

Il reste encore un dernier point à observer : à chaque fois que la joueuse $\exists$ choisit la formule $F \to \bot$ (c'est-à-dire à chaque fois que le client demande une réinitialisation de la session), $\forall$ choisit à nouveau le nombre de paquets qu'il désire envoyer, sans être contraint de toujours choisir le même.

4.3.3 Bonne fondation et initialisation de la session

La procédure d'initialisation

Nous allons décrire dans cette partie, toujours grâce au jeu associé à une formule bien choisie, la procédure d'initialisation dite du *three-way handshake*, au cours de laquelle le client et le serveur doivent échanger leurs numéros de séquence initiaux.
Les protocoles réseaux n'interdisant pas aux connexions de s'ouvrir et de se fermer aussi fréquemment et rapidement que possible, les numéros de séquence contenus dans un message doivent permettre aux protagonistes d'identifier à quelle session il appartient, et donc *in fine* au client de recomposer les données sans erreur.
L'initialisation nécessite l'envoi d'accusés de réception dans les deux sens, et ceux-ci sont interdépendants : chacun propose à l'autre un numéro de séquence initial qui lui permettra de distinguer cette session des autres ; le couple formé par les deux numéros définissant la connexion, il apparaitra dans l'en-tête de chaque message envoyé ultérieurement.
Il semble que si le joueur $\forall$ est obligé d'envoyer un acquittement pour que la session soit couronnée de succès, la formule décrivant ce protocole n'est pas ω-valide puisque le joueur $\forall$ possède une stratégie gagnante : ne jamais envoyer d'accusé de réception. Néanmoins, la nature particulière de la procédure d'initialisation fait qu'il n'en est rien.
Remarquons déjà que la demande d'ouverture de session peut être faite par l'un ou l'autre des deux partis. Dans le jeu décrit ci-dessous, le joueur $\forall$ représente toujours l'expéditeur du premier message de synchronisation (celui qui cherche à établir la connexion), il peut donc s'agir indifféremment du client ou du serveur, alors que la joueuse $\exists$ représente l'ensemble formé par la ligne et le destinataire de ce message. Une session de communication complète sera donc modélisée par deux formules, c'est-à-dire par deux jeux successifs :

- le premier décrivant la procédure d'initialisation, où $\forall$ représente celui qui initie la synchronisation.
- le second décrivant l'envoi et l'acquittement des données, où $\forall$ représente le serveur et $\exists$ le client : les rôles des joueurs sont éventuellement intervertis. La formule décrivant ce jeu fera apparaître explicitement les numéros de séquence choisis à la première partie.

Le passage d'un jeu à un autre permet également de vider l'ensemble $\mathscr{U}$ des formules qu'il contient, ce qui modélise le fait que les messages envoyés au cours de la synchronisation ne peuvent pas interférer avec les paquets de données une fois la connexion établie.

On considère la formule suivante, où ψ est une fonction récursive telle que la relation $\psi(m,n,p,q) = 1$ définisse une relation $(p,q)R(m,n)$ bien fondée sur $\mathbb{N}^2$:

$$\forall X \forall n \forall p \left(\forall n' \forall p' \left(\forall n'' \forall p'' \{ \psi(n',p',n'',p'') = 1 \rightarrow Xn'',p'' \} \rightarrow Xn',p' \right) \rightarrow Xn,p \right).$$

Si l'on choisit pour X un prédicat P ne dépendant que de son premier argument, et pour ψ une fonction ne dépendant pas de son troisième argument, on obtient la formule ω-valide suivante :

$$\forall n \left(\forall n' \forall p' \left(\forall p'' \{ \psi(n',p',p'') = 1 \rightarrow Pp'' \} \rightarrow Pn' \right) \rightarrow Pn \right).$$

Nous allons dans la suite considérer la formule

$$\forall n \left(\forall n' \forall p \left(\forall q \{ \psi(n',p,q) = 1 \rightarrow Pq \} \rightarrow Pn' \right) \rightarrow Pn \right),$$

où ψ est définie de la manière suivante :

$\psi(n,p,q) = 1$ si, et seulement si, n est impair et $q = 2p+(n+p)(n+p+1)$.

Pour tout entier p_0, la relation définie par $\psi(n,p_0,q) = 1$ est bien fondée puisque toute suite décroissante maximale est de longueur égale à 2, car n doit être impair et q pair pour que ψ puisse prendre la valeur 1.
Il faut également noter que $\psi(n,p,q) = 1$ implique que q est un code pour (n,p). On posera donc dans la suite $< n,p >= 2p+(n+p)(n+p+1)$

Intuitivement, le jeu associé à cette formule est le suivant :

1. Le joueur $\forall$ cherche à établir la connexion : il choisit son numéro de séquence n qu'il envoie à la joueuse $\exists$.
2. La joueuse $\exists$ reçoit alors ce message et choisit son numéro de séquence initial p. Elle envoie alors un message contenant les entiers n et p dans son en-tête. Ce message acquitte donc celui de $\forall$ grâce à la présence de n, et appelle à son tour un acquittement de p.
3. Enfin, $\forall$ doit choisir un entier q qui code n et p, qui sera considéré comme l'acquittement de p : il s'agit du numéro définissant la session en cours d'initialisation. Il envoie alors cette donnée au destinataire : la partie se termine et l'envoi des données peut débuter.

Dans ce qui suit, on suppose dans un premier temps que **c'est le client qui cherche à établir la connexion** : il est donc représenté dans la partie par le joueur $\forall$, alors que la joueuse $\exists$ correspond à l'ensemble formé par le serveur et la ligne. La procédure est donc décrite du point de vue du client.

Représentons déjà la partie la plus simple possible.
On pose $G(n',p) = \forall q \{ \psi(n',p,q) = 1 \rightarrow Pq \} \rightarrow Pn'$.

$\mathscr{U}$	$\mathscr{V}$	$\mathscr{A}$	Action
$F \rightarrow \bot$	F	$\bot$	$\forall$ choisit F et n_1 impair
$\forall n' \forall p\ G(n', p)$	inchangé	Pn_1	$\exists$ choisit $\forall n' \forall p\ G(n', p)$, $n' = n_1$ et p_1
inchangé	$\forall q\{\psi(n_1, p_1, q) = 1 \rightarrow Pq\}$	inchangé	$\forall$ choisit $\forall q\{\psi(n_1, p_1, q) = 1 \rightarrow Pq\}$ et $q_1 = < n_1, p_1 >$
inchangé	inchangé	Pq_1	$\exists$ choisit $\forall n' \forall p\ G(n', p)$, $n' = q_1$ et p_2
inchangé	$\forall q\{\psi(q_1, p_2, q) = 1 \rightarrow Pq\}$	inchangé	$\forall$ ne peut choisir q tel que $\psi(q_1, p_2, n) = 1$, il perd donc la partie.

Ce tableau s'interprète d'une manière sensiblement différente de ceux vus précédemment. Interprétons-le ligne par ligne :

1ière ligne : Le client envoie une requête au serveur, contenant son numéro de séquence initial n_1.

2ième ligne : $\exists$ modélise :

1. un phénomène se produisant sur le canal : la requête est arrivée jusqu'au serveur,
2. le comportement du serveur qui acquitte le message du client en renvoyant $n' = n_1$ ainsi que son propre numéro de séquence initial p_1,
3. un phénomène se produisant sur le canal : ce message est arrivé jusqu'au client.

3ième ligne : $\forall$ modélise par le choix de $q_1 = < n_1, p_1 >$ le comportement du client, qui renvoie q_1 codant le couple constitué par les numéros choisis.

4ième ligne : $\exists$ modélise uniquement un phénomène ayant lieu sur le canal de communication : le serveur a bien reçu le dernier message du client, qui contient le numéro q_1 de la session qui peut maintenant débuter. Mais on ne considère pas ici que le prédicat Pq_1 représente un message envoyé par le serveur.

5ième ligne : Comme $\forall$ perd la partie quel que soit son choix, on considère que cette ligne ne possède pas d'interprétation du point de vue du protocole.

On considère que la session s'arrête à la 4ième ligne, puisque $\exists$ est certaine d'emporter la partie, et qu'à cet instant le serveur n'envoie aucun message.

Si à la deuxième ligne la joueuse $\exists$ choisit la formule $F \to \bot$, on considère que le serveur n'a pas reçu le message du client, et que celui-ci le renvoie à la fin du timeout.
Si le joueur $\forall$ choisit à la première ligne un entier n pair, il perd fatalement la partie lorsqu'il doit choisir la formule $\forall q\{\psi(n, p_1, q) = 1 \to Pq\}$, puisqu'il ne peut en satisfaire l'hypothèse. On considère dans ce cas que la requête de connexion n'était pas valide, et que le serveur a ensuite envoyé un message lui indiquant qu'il refusait d'établir la connexion. Le client devra alors reprendre la procédure du début.
Modéliser le comportement du client par le joueur $\forall$ modélise la contrainte qui pèse sur lui au moment de l'initialisation : il est toujours obligé de répondre au dernier message reçu. Il est ainsi contraint de tenir compte des acquittements qu'il reçoit, et de poursuivre la procédure en considérant le dernier numéro initial reçu. Ainsi, à la troisième ligne, $\forall$ est obligé de choisir le numéro q_1, sous peine de perdre la partie. Le protocole spécifie donc qu'une fois que le client a amorcé l'initialisation, il est obligé de la terminer le plus rapidement possible, sous peine de devoir reprendre la procédure du début. Cela permet de prévenir le blocage du serveur par des clients paresseux, qui amorcent une initialisation et tardent ensuite à la clore.

Considérons maintenant la partie suivante.

$\mathscr{U}$	$\mathscr{V}$	$\mathscr{A}$	Action
$F \to \bot$	F	$\bot$	$\forall$ choisit F et n_1 impair
$\forall n' \forall p\ G(n', p)$	inchangé	Pn_1	$\exists$ choisit $\forall n' \forall p\ G(n', p)$, $n' = n_1$ et p_1
inchangé	$\forall q\{\psi(n_1, p_1, q) = 1 \to Pq\}$	inchangé	$\forall$ choisit $\forall q\{\psi(n_1, p_1, q) = 1 \to Pq\}$ et $q_1 =< n_1, p_1 >$
inchangé	inchangé	Pq_1	$\exists$ choisit $F \to \bot$
inchangé	F	inchangé	$\forall$ choisit F et n_2 impair
inchangé	inchangé	Pn_2	$\exists$ choisit $\forall n' \forall p\ G(n', p)$, $n' = n_2$ et p_2
inchangé	$\forall q\{\psi(n_2, p_2, q) = 1 \to Pq\}$	inchangé	$\forall$ choisit $\forall q\{\psi(n_2, p_2, q) = 1 \to Pq\}$ et $q_2 =< n_2, p_2 >$
inchangé	inchangé	Pq_2	$\exists$ choisit $\forall n' \forall p G(n', p)$, $n' = q_1$ et p_3
inchangé	$\forall q\{\psi(q_1, p_3, q) = 1 \to Pq\}$	inchangé	$\forall$ ne peut choisir q tel que $\psi(q_1, p_3, q) = 1$, il perd donc la partie.

A la cinquième ligne de ce tableau, le client envoie à nouveau un premier message de synchronisation, puisqu'il n'a pas reçu d'acquittement du serveur avant la fin du timeout. A l'avant-dernière ligne, $\exists$ modélise la réception par le serveur du message envoyé à la troisième : la synchronisation est terminée. A la fin de cette partie, le client n'a aucun moyen de savoir lequel de ses deux acquittements est arrivé jusqu'au serveur, il ne sait donc pas lequel des deux numéros q_1, q_2 sera associé à la session qui s'amorce (il ne sait d'ailleurs même pas que l'initialisation a été couronnée de succès). C'est le premier paquet de données qu'il recevra qui le renseignera sur ce fait.

Dans le cas où c'est le serveur qui initie la procédure de connexion, le client est modélisé par la joueuse $\exists$, et le protocole lui impose moins de contraintes : si le serveur initie la session, c'est qu'il est certain de la fiabilité de son interlocuteur, et le protocole n'a plus besoin d'assurer la sécurité décrite plus haut. On doit alors modifier certaines des interprétations précédentes. En l'occurrence, on doit considérer que dans le premier de ces deux tableaux :

- si le joueur $\forall$ (qui représente maintenant le serveur) choisit un individu $q_1 \neq < n_1, p_1 >$, à la troisième ligne, on considère qu'il s'agit d'un déni de service classique.
- à la quatrième ligne, le comportement de la joueuse doit être réinterprété : en effet, le serveur n'attend pas que le client ait reçu le dernier message de synchronisation avant d'envoyer les données. L'interprétation est alors la suivante :
  - si elle choisit de terminer la partie, cela modélise le fait qu'aucun message de réinitialisation ou de synchronisation n'est arrivé au serveur avant que celui-ci décide d'envoyer le premier paquet de données : la partie se termine et l'envoi des données peut ensuite commencer ;
  - si à cet instant elle décide par le choix d'une autre formule de revenir en arrière, on considère que le client a renvoyé un message de réinitialisation ou de synchronisation, et que le serveur l'a reçu avant de transmettre le premier paquet de données : la procédure d'initialisation doit alors reprendre (cette situation est alors décrite par le second tableau).

L'envoi des données après l'initialisation

On considère une partie associée à la formule définie précédemment, et remportée par $\exists$ en choisissant lors de son dernier tour un entier q.
On peut décrire l'envoi des paquets de données suivant cette initialisation par la formule suivante :

$$\forall n \forall x \left(\forall p \forall x' \left(\forall m \forall y \{ sm = p \rightarrow Pq, m, y \} \rightarrow Pq, p, x' \right) \rightarrow Pq, n, x \right)$$

L'entier q est alors le numéro de la session en cours : il apparaît dans l'en-tête de chaque message envoyé au cours de cette session, et contient les deux nu-

méros de séquence initiaux. Le premier paquet de données indiquera donc au client quel est le numéro de la session qui débute avec ce message.

Il est à noter que ce protocole gère les numéros de séquence différemment de TCP. Là où TCP ne considère qu'un entier, qui est la somme du numéro d'ordre du paquet de données et du numéro initial, le protocole décrit ici en emploie deux : le couple formé par le numéro d'ordre du paquet de données et l'entier définissant la session. Au moment de la conception de TCP les bandes passantes étaient en effet beaucoup plus étroites qu'aujourd'hui, et il fallait donc limiter au maximum la taille de l'en-tête. La procédure de connexion décrite ici a l'avantage d'être moins lourde à mettre en place : il n'est en effet pas nécessaire que les numéros initiaux choisis *in fine* soit les plus grands parmi ceux qui ont été proposés.

Chapitre 5

Les jeux en réalisabilité

Dans la section 2.2, nous avons vu qu'un terme réalisant la restriction aux entiers d'une formule de la forme $\exists x \forall y f(x,y) = 0$ implémentait une stratégie gagnante pour la joueuse $\exists$. Nous allons voir dans ce chapitre qu'un réalisateur d'une formule valide implémente une famille de parties dans le jeu associé. Cela permettra de donner des processus dont l'exécution décrit des sessions de communication entre deux machines suivant les protocoles donnés dans le chapitre précédent.

5.1 Formules normales

On considère dans cette section les formules usuelles écrites sans quantificateur du second ordre, mais éventuellement avec les symboles $\bot$ et $=$. On considère les expressions de la forme $\tau_1 = \tau_2$ comme des formules atomiques. Les termes considérés sont ceux donnés dans la définition 2.1.11.

Définition 5.1.1.
Une formule est dite « sous forme normale » ou simplement « normale » si on peut l'obtenir par les règles suivantes :
- *toute formule atomique est normale ;*
- *si $\Phi_1, \dots, \Phi_p$ sont des formules normales, si A est atomique et si $x_1, \dots, x_k$ sont des variables d'individus, alors $\forall x_1 \dots \forall x_k (\Phi_1, \dots, \Phi_p \to A)$ est normale.*

Nous allons voir que l'on peut associer à chaque formule du premier ordre une formule normale, de sorte que les deux formules aient la même valeur de vérité.

Notations : Si $p = 0$, l'expression $\forall x_1 \dots \forall x_k (\Phi_1, \dots, \Phi_p \to A)$ dénotera la formule $\forall x_1 \dots \forall x_k \, A$.
Si $k = 0$, $\forall x_1 \dots \forall x_k (\Phi_1, \dots, \Phi_p \to A)$ dénotera la formule $\Phi_1, \dots, \Phi_p \to A$. Il en résulte que toute formule normale est de la forme $\forall x_1 \dots \forall x_k (\Phi_1, \dots, \Phi_p \to A)$.

Définition 5.1.2.
A chaque formule F, on peut associer une formule normale notée $\tilde{F}$ et appelée « forme normale de F », telle que :
- *si F est atomique, alors $\tilde{F} = F$;*
- *si F est de la forme $\forall x\ G$, alors $\tilde{F} = \forall x\ \tilde{G}$;*
- *si F est de la forme $G \to H$, et si $\tilde{H} = \forall x_1 \dots \forall x_k(\Phi_1, \dots, \Phi_p \to A)$ de sorte que les x_i ne sont pas libres dans G, alors $\tilde{F} = \forall x_1 \dots \forall x_k(\tilde{G}, \Phi_1, \dots, \Phi_p \to A)$.*

Par exemple, la forme normale de $\forall x Px \to \forall x Px$ est $\forall y(\forall x Px \to Py)$.

Théorème 5.1.3.
Soit F une formule. Alors F et $\tilde{F}$ sont équivalentes en logique classique.

Démonstration. La preuve se fait par induction sur la formule. La seule difficulté se présente dans le cas où F s'écrit $G \to H$; il suffit alors de vérifier que les formules $G \to \forall x_1 \dots \forall x_k(\Phi_1, \dots, \Phi_p \to A)$ et $\forall x_1 \dots \forall x_k(G, \Phi_1, \dots, \Phi_p \to A)$ sont équivalentes si les x_i ne sont pas libres dans G, ce qui est trivial. □

Le point important est le théorème suivant, qui montre que la mise sous forme normale d'une formule close ne modifie pas sa valeur de vérité : on peut donc ne considérer que des formules normales.

Théorème 5.1.4.
Soit F une formule close. Alors $||F|| = ||\tilde{F}||$.

Démonstration. On procède par induction sur F. Le seul cas non-trivial est celui où F s'écrit $G \to H$. Si $\forall x_1 \dots \forall x_k(\Phi_1, \dots, \Phi_p \to A)$ est la forme normale de H, on a alors

$$\begin{aligned}
||\tilde{F}|| &= ||\forall x_1 \dots \forall x_k(\tilde{G}, \Phi_1, \dots, \Phi_p \to A)|| \text{ par définition de } \tilde{F} \\
&= ||\tilde{G} \to \forall x_1 \dots \forall x_k(\Phi_1, \dots, \Phi_p \to A)|| \text{ en utilisant le lemme 2.1.20} \\
&= ||\tilde{G} \to \tilde{H}|| \\
&= ||G \to H|| \text{ par l'hypothèse de récurrence et le lemme 2.1.24} \\
&= ||F||,
\end{aligned}$$

ce qui donne le résultat. □

5.2 Spécification des formules valides à l'aide de jeux

Il n'est pas question dans cette section d'ω-langage. On jouera sur les formules normales écrites sans le symbole $=$. Pour une généralisation des résultats de ce chapitre, on se reportera à [17] et [9].

Soit $\mathscr{N}$ un ensemble **dénombrable**, à partir duquel on développe la théorie de la réalisabilité classique. On jouera dans cette section sur des formules à paramètres dans $\mathscr{N}$. On se donne encore des symboles de prédicat en nombre dénombrable notés $R, P, \ldots$

Définition 5.2.1.
On appelle formule jouable toute formule normale close écrite sans le symbole $=$.

Considérant un symbole de prédicat R, on associe une constante de pile notée $\pi_{Ri_1,\ldots,i_k}$ à chaque formule close à paramètres de la forme $Ri_1, \ldots, i_k$, ce qui permet d'étendre de manière naturelle la notion de valeur de vérité aux formules écrites avec des symboles de prédicat.

A chaque formule jouable Φ à paramètres, on associe une instruction notée κ_Φ. Sa règle d'exécution sera non déterministe puisqu'elle dépendra du comportement des joueurs $\forall$ et $\exists$.

On commence par étendre la notion de valeur de vérité aux formules jouables, et par associer à chacune d'entre elles un second ensemble de pile noté $[\![\Phi]\!]$.

Définition 5.2.2.
Soit Φ une formule jouable. On définit deux ensembles de piles notés $||\Phi||$ et $[\![\Phi]\!]$ avec les règles suivantes :

- *Si $\Phi = \bot$, on pose $[\![\Phi]\!] = ||\Phi|| = \Pi$.*
- *Si $\Phi = Ri_1, \ldots, i_k$, on pose $||\Phi|| = [\![\Phi]\!] = \{\pi_\Phi\}$.*
- *Si $\Phi = \forall x_1 \ldots \forall x_k(\Psi_1(x_1, \ldots, x_k), \ldots, \Psi_p(x_1, \ldots, x_k) \rightarrow Ax_1, \ldots, x_k)$, on pose*
  $||\Phi|| = \bigcup_{i_1,\ldots,i_k \in \mathscr{N}} \{t_1 \cdot \ldots \cdot t_p \cdot \pi;\ t_j \Vdash \Psi_j(i_1, \ldots, i_k), \pi \in ||Ai_1, \ldots, i_k||\}$
  et $[\![\Phi]\!] = \bigcup_{i_1,\ldots,i_k \in \mathscr{N}} \{\kappa_{\Psi_1(i_1,\ldots,i_k)} \cdot \ldots \cdot \kappa_{\Psi_p(i_1,\ldots,i_k)} \cdot \pi;\ \pi \in [\![Ai_1, \ldots, i_k]\!]\}$.

On peut alors définir la règle d'exécution d'une instruction κ_Φ, où Φ s'écrit $\forall x_1 \ldots \forall x_k(\Psi_1(x_1,\ldots,x_k),\ldots,\Psi_p(x_1,\ldots,x_k) \to Ax_1,\ldots,x_k)$, de la manière suivante :

$$\kappa_\Phi \star \xi_1 \cdot \ldots \cdot \xi_p \cdot \pi \succ \xi_j \star \rho$$

où $j \in \{1,\ldots,p\}$ et $\rho \in \Pi$ sont définis comme suit :

1. $\exists$ choisit d'abord $i_1,\ldots,i_k$ dans $\mathscr{N}$ de sorte que $\pi \in [\![Ai_1,\ldots,i_k]\!]$.
   Si elle ne peut satisfaire cette condition, l'exécution s'arrête sur le processus $\kappa_\Phi \star \xi_1 \cdot \ldots \cdot \xi_p \cdot \pi$.
2. $\forall$ choisit alors $j \in \{1,\ldots,p\}$ et une pile $\rho \in [\![\Psi_j(i_1,\ldots,i_k)]\!]$. S'il ne peut le faire, c'est-à-dire si la formule Φ est la clôture d'une formule atomique, on considère que l'exécution s'arrête sur le processus $\kappa_\Phi \star \xi_1 \cdot \ldots \cdot \xi_p \cdot \pi$.

Chaque instruction κ_Φ est interactive et permet aux joueurs d'agir sur l'exécution. On peut alors associer un jeu à tout processus p : chaque exécution correspondra à une partie dans ce jeu. On considère que $\exists$ remporte une partie donnée si et seulement si l'exécution correspondante se termine sur un processus de la forme $\kappa_\Phi \star \pi$ avec $\pi \in [\![\Phi]\!]$, où Φ est la clôture d'une formule atomique. C'est-à-dire que la joueuse l'emporte si $\forall$ ne peut plus jouer.

Considérant la définition de $[\![\Psi_j(i_1,\ldots,i_k)]\!]$, on voit que le choix d'une pile $\rho \in [\![\Psi_j(i_1,\ldots,i_k)]\!]$ par le joueur $\forall$ correspond à une instanciation de cette formule. Pour une formule jouable donnée, ces règles sont donc les mêmes que celles définies dans le chapitre 4, à ceci près que le processus considéré effectue maintenant les choix de formules dans l'ensemble $\mathscr{U}$ à la place de la joueuse.

Lemme 5.2.3.
Soit $\perp\!\!\!\perp = \{p \in \Lambda_c \star \Pi; \exists$ *possède une stratégie gagnante pour le jeu associé à* $p\}$.
Alors quelle que soit la formule jouable Φ, *on a* $[\![\Phi]\!] \subset ||\Phi||$ *et* $\kappa_\Phi \Vdash \Phi$.

Démonstration. L'ensemble $\perp\!\!\!\perp$ défini ici est clairement saturé pour les règles d'exécution données à la définition 2.1.9. Montrons le résultat par induction sur la formule Φ choisie.
Si Φ est atomique, le résultat est trivial.
Sinon, on prend $\Phi = \forall x_1 \ldots \forall x_k(\Psi_1(x_1,\ldots,x_k),\ldots,\Psi_p(x_1,\ldots,x_k) \to Ax_1,\ldots,x_k)$, et $i_1,\ldots i_k$ dans $\mathscr{N}$. Par l'hypothèse d'induction on a $[\![Ai_1,\ldots,i_k]\!] \subset ||Ai_1,\ldots,i_k||$ et $\kappa_{\Psi_j(i_1,\ldots,i_k)} \Vdash \Psi_j(i_1,\ldots,i_k)$, ce qui assure $[\![\Phi]\!] \subset ||\Phi||$. Considérons maintenant des termes $\xi_1,\ldots,\xi_p$ tels que ξ_j réalise $\Psi_j(i_1,\ldots,i_k)$, et $\pi \in ||Ai_1,\ldots,i_k||$. On doit montrer $\kappa_\Phi \star \xi_1 \cdot \ldots \cdot \xi_p \cdot \pi \in \perp\!\!\!\perp$, c'est-à-dire que $\exists$ possède une stratégie gagnante pour le jeu associé à ce processus. La stratégie consiste à d'abord choisir les individus $i_1,\ldots,i_k$; on a en effet $\pi \in [\![Ai_1,\ldots,i_k]\!]$ puisque A est atomique. Le joueur $\forall$ choisit alors $j \in \{1,\ldots,p\}$ tel que $\Psi_j(i_1,\ldots,i_k)$ est une formule jouable et $\rho \in [\![\Psi_j(i_1,\ldots,i_k)]\!]$. S'il ne peut le faire, $\exists$ remporte la partie, et donc cette stratégie est gagnante ; sinon le processus considéré se réduit en $\xi_j \star \rho$. Mais,

toujours par l'hypothèse d'induction, il vient $\rho \in ||\Psi_j(i_1, \ldots, i_k)||$, et donc que ce processus est dans $\perp\!\!\!\perp$ puisque ξ_j réalise cette formule. La joueuse n'a donc plus qu'à appliquer une stratégie gagnante associée au processus $\xi_j \star \rho$. □

Théorème 5.2.4.
Si Φ est une formule jouable et si $\theta \Vdash \Phi$, alors pour toute pile $\pi \in [\![\Phi]\!]$, la joueuse $\exists$ possède une stratégie gagnante pour le jeu associé au processus $\theta \star \pi$.

Démonstration. On prend

$$\perp\!\!\!\perp = \{p \in \Lambda_c \star \Pi; \exists \text{ possède une stratégie gagnante pour le jeu associé à } p\};$$

le lemme 5.2.3 assure alors $\pi \in ||\Phi||$, ce qui donne le résultat. □

Soit F une formule jouable, considérons un processus de la forme $\theta \star \pi$, où $\theta \Vdash \neg\neg F$ et $\pi \in [\![\neg\neg F]\!]$. Les parties associées à celui-ci correspondent à une famille de parties associées à la formule F dans le jeu défini en 4.2. On considère $\neg\neg F$ au lieu de F afin de permettre à la joueuse $\exists$ de réinitialiser la partie. En effet, si F commence par un quantificateur universel, le choix d'une pile $\pi \in [\![F]\!]$ correspond à une instanciation de cette formule. Si F ne commence pas par un quantificateur universel, on peut par contre considérer les jeux associés aux processus de la forme $\xi \star \rho$, avec $\xi \Vdash F$ et $\rho \in [\![F]\!]$.

Un tour de jeu a lieu chaque fois qu'une constante κ_Φ arrive en tête. On a alors affaire à un processus de la forme $\kappa_\Phi \star \xi_1 \cdot \ldots \cdot \xi_p \cdot \pi_A$ et le processus a déjà choisi les formules Φ et A à la place de la joueuse $\exists$. Celle-ci n'a plus qu'à choisir les individus $i_1, \ldots, i_k$. Le joueur $\forall$ choisit alors un individu j et une pile $\rho = \kappa_{\Phi'_1(j_1,\ldots,j_q)} \cdot \ldots \cdot \kappa_{\Phi'_n(j_1,\ldots,j_q)} \cdot \pi' \in [\![\Psi_j(j_1, \ldots, j_q)]\!]$, c'est-à-dire que cette formule est instanciée par l'introduction de nouvelles constantes d'interaction et d'une nouvelle pile. Le jeu se poursuit ensuite avec le processus $\xi_j \star \rho$.

Le théorème 5.2.4 donne ainsi une mise en œuvre partielle du théorème 4.2.10, et permet à partir d'une preuve d'une formule valide d'implémenter une partie dans laquelle la joueuse $\exists$ possède une stratégie gagnante. Cela permet de classifier les preuves d'une formule valide, et plus généralement les termes qui la réalisent, en fonction des stratégies gagnantes qu'ils induisent.

Néanmoins, toutes les stratégies gagnantes pour $\exists$ dans le jeu associé à F ne sont pas représentées par des quasi-preuves. Considérons par exemple la formule valide suivante :

$$\forall x(\forall y Ry, R0 \to Rx).$$

La stratégie de $\exists$ consistant à choisir $R0$ si $\forall$ a choisi $x = 0$ au premier tour, $\forall y Ry$ et y=0 sinon, est gagnante mais n'est pas représentable par une quasi-preuve. En effet, l'individu x choisi par $\forall$ n'apparaîtra dans un processus qu'en

tant qu'indice de l'instruction κ_{R0}, et est ainsi inaccessible à une quasi-preuve, qui ne pourra donc pas s'exécuter en fonction de celui-ci.

Remarque : On peut également étendre cette notion de jeu aux formules dont les quantificateurs sont restreints aux entiers (Cf. [9]) ; dans ce cas, chaque quasi-preuve réalisant une formule F dont les quantificateurs sont restreints implémente une stratégie gagnante pour la joueuse $\exists$.

5.3 Quelques exemples

5.3.1 Envoi d'un paquet

On considère ici la formule $\neg\neg\forall x(\forall y Py \rightarrow Px)$. On note κ la constante d'interaction associée à la formule $\neg\forall x(\forall y Py \rightarrow Px)$. Elle possède la règle d'exécution suivante :

$$\kappa \star \xi \cdot \pi \succ \xi \star \kappa_{\forall y Py} \cdot \pi_{Pj}$$

où l'individu j est choisi par le joueur $\forall$.

L'instruction $\kappa_{\forall y Py}$ ne possède elle aucune règle d'exécution et :
- pour tout individu j, le processus $\kappa_{\forall y Py} \star \pi_{Pj}$ est gagnant pour la joueuse : celle-ci est obligé de choisir j et $\forall$ ne peut plus faire aucun choix.
- si la pile ρ n'est associée à aucune formule atomique Pj, le processus $\kappa_{\forall y Py} \star \rho$ est gagnant pour le joueur $\forall$: $\exists$ ne peut trouver j tel que $\rho \in [\![Pj]\!] = \{\pi_{Pj}\}$.

Considérant un processus p contenant κ comme seule instruction interactive, la joueuse $\exists$ l'emporte donc si et seulement si sa réduction se termine sur un processus de la forme $\kappa_{\forall y Py} \star \pi_{Pj}$.

On considère que l'arrivée en tête de l'instruction κ correspond à l'envoi par le serveur d'un message, celui-ci étant représenté par la pile π_{Pj}. L'arrivé de l'instruction $\kappa_{\forall y Py}$ en tête modélise la fin de la session, c'est-à-dire l'arrivée d'un acquittement ; la session est couronnée de succès ssi la pile courante est de la forme π_{Pj}, et on considère alors que les données acquittées possédaient l'en-tête j.

Théorème 5.3.1.
La formule $\forall P\{\neg\neg\forall x(\forall y Py \rightarrow Px)\}$ est réalisée par les quasi-preuves suivantes :
i) $\lambda u(u)I$.
ii) $\lambda u(u)(u)I$.
iii) $\lambda u(u)\lambda g(\mathsf{cc})u$.

Démonstration. On note U le contexte $u : \forall x(\forall y Py \to Px) \to \bot$. Le résultat découle du lemme d'adéquation appliqué aux preuves suivantes.

i)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{z : \forall y Py \vdash z : \forall y Py}}{z : \forall y Py \vdash z : Px}}{\vdash \lambda z\, z : \forall y Py \to Px}}{\vdash \lambda z\, z : \forall x(\forall y Py \to Px)} \quad \dfrac{}{U \vdash u : \forall x(\forall y Py \to Px) \to \bot}}{U \vdash (u)\lambda z\, z : \bot}}{\vdash \lambda u(u)\lambda z\, z : [\forall x(\forall y Py \to Px) \to \bot] \to \bot}}{\vdash \lambda u(u)\lambda z\, z : \forall P([\forall x(\forall y Py \to Px) \to \bot] \to \bot)}$$

ii)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{z : \forall y Py \vdash z : \forall y Py}}{z : \forall y Py \vdash z : Px}}{\vdash \lambda z\, z : \forall y Py \to Px}}{\vdash \lambda z\, z : \forall x(\forall y Py \to Px)} \quad \dfrac{}{U \vdash u : \forall x(\forall y Py \to Px) \to \bot}}{U \vdash (u)\lambda z\, z : \bot}}{U \vdash (u)\lambda z\, z : \forall x(\forall y Py \to Px)} \quad \dfrac{}{U \vdash u : \forall x(\forall y Py \to Px) \to \bot}}{U \vdash (u)(u)\lambda z\, z : \bot}}{\vdash \lambda u(u)(u)\lambda z\, z : [\forall x(\forall y Py \to Px) \to \bot] \to \bot}}{\vdash \lambda u(u)(u)\lambda z\, z : \forall P([\forall x(\forall y Py \to Px) \to \bot] \to \bot)}$$

iii)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{g : \forall y Py \vdash u : \forall x(\forall y Py \to Px) \to \bot}}{g : \forall y Py \vdash u : (\forall y Py \to \forall x Px) \to \forall y Py}\,\mathrm{R}}{g : \forall y Py \vdash (\mathsf{cc})u : \forall y Py}}{g : \forall y Py \vdash (\mathsf{cc})u : Px}}{\vdash \lambda g(\mathsf{cc})u : \forall y Py \to Px}}{\vdash \lambda g(\mathsf{cc})u : \forall x(\forall y Py \to Px)} \quad \dfrac{}{U \vdash u : \forall x(\forall y Py \to Px) \to \bot}}{U \vdash (u)\lambda g(\mathsf{cc})u : \bot}}{\vdash \lambda u(u)\lambda g(\mathsf{cc})u : [\forall x(\forall y Py \to Px) \to \bot] \to \bot}}{\vdash \lambda u(u)\lambda g(\mathsf{cc})u : \forall P([\forall x(\forall y Py \to Px) \to \bot] \to \bot)}$$

□

Décrivons les exécutions associées à ces termes, en notant $j_1, \ldots, j_k$ les individus successivement joués par $\forall$ au cours d'une partie.

$$\begin{aligned}\lambda u(u)I \star \kappa \cdot \pi &\succ \kappa \star I \cdot \pi \\ &\succ I \star \kappa_{\forall y P y} \cdot \pi_{P j_1} \\ &\succ \kappa_{\forall y P y} \star \pi_{P j_1}\end{aligned}$$

La stratégie gagnante pour la joueuse est dans ce cas de choisir $y = i_1$, ce qu'elle est d'ailleurs contrainte (par les règles du jeu) de faire. Ce terme correspond donc à une session où le paquet de données n'a été envoyé qu'une seule fois, et acquitté de suite.

$$\begin{aligned}\lambda u(u)(u)I \star \kappa \cdot \pi &\succ \kappa \star (\kappa)I \cdot \pi \\ &\succ \kappa \star I \cdot \kappa_{\forall y P y} \cdot \pi_{P j_1} \\ &\succ I \star \kappa_{\forall y P y} \cdot \pi_{P j_2} \\ &\succ \kappa_{\forall y P y} \star \pi_{P j_2}\end{aligned}$$

La stratégie gagnante pour la joueuse est dans ce cas de choisir $y = j_2$. Ce terme correspond donc à une session où les données ont été envoyées deux fois, et où le second paquet a été acquitté.

$$\begin{aligned}\lambda u(u)\lambda g(\mathsf{cc})u \star \kappa \cdot \pi &\succ \kappa \star \lambda g(\mathsf{cc})\kappa \cdot \pi \\ &\succ \lambda g(\mathsf{cc})\kappa \star \kappa_{\forall y P y} \cdot \pi_{P j_1} \\ &\succ \mathsf{cc} \star \kappa \cdot \pi_{P j_1} \\ &\succ \kappa \star \mathsf{k}_{\pi_{P j_1}} \cdot \pi_{P j_1} \\ &\succ \mathsf{k}_{\pi_{P j_1}} \star \kappa_{\forall y P y} \cdot \pi_{P j_2} \\ &\succ \kappa_{\forall y P y} \star \pi_{P j_1}\end{aligned}$$

La stratégie gagnante pour la joueuse est dans ce cas de choisir $y = j_1$. Ce terme correspond donc à une session où le message a été envoyé deux fois avant d'être acquitté.

On voit sur ces exemples, qu'à un réalisateur de ce théorème correspond une unique stratégie gagnante pour $\exists$, celui-ci décrivant une session d'un type bien précis : le message et l'acquittement qui mettent un terme à la session sont déterminés de façon univoque.

5.3.2 Le buveur

Dans l'exemple précédent, la joueuse $\exists$ n'a aucun rôle puisqu'elle est contrainte par les règles du jeu d'appliquer la stratégie gagnante induite par la quasi-preuve considérée. Les réalisateurs de la formule considérée dans cette section lui laissent par contre la liberté d'employer une stratégie gagnante ou pas.
On considère la formule $\exists x \forall y(Rx \to Ry)$. Cette formule est valide puisqu'il suffit de définir x comme étant un individu tel que $\neg R(x)$ s'il en existe un, tout x convenant sinon.
La constante d'interaction associée à la formule $\forall x(\forall y(Rx \to Ry) \to \bot)$ sera notée κ, et possède la règle d'exécution suivante :

$$\kappa \star \xi \cdot \pi \succ \xi \star \kappa_{Ri} \cdot \pi_{Rj}$$

où l'individu i est choisi par le joueur $\exists$, l'individu j par l'opposant $\forall$. Considérant un processus p, le joueur $\exists$ l'emporte dans la partie associée à celui-ci si l'exécution se termine sur un processus de la forme $\kappa_{Ri} \star \pi_{Ri}$. Il est clair qu'une quasi-preuve implémentant une stratégie gagnante pour $\exists$ utilise l'instruction cc, ce qui exprime le fait que la formule considérée n'est pas démontrable en logique intuitionniste.

Théorème 5.3.2.
La formule $\forall R\{\forall x(\forall y(Rx \to Ry) \to \bot) \to \bot\}$ est réalisée par les quasi-preuves suivantes :

i) $\lambda z(z)\lambda g(\mathsf{cc})z$;
ii) $\lambda z(z)\lambda g(\mathsf{cc})\lambda k(z)(z)k$.

Démonstration.
i) Notons Z le contexte $z : \forall x(\forall y(Rx \to Ry) \to \bot)$, et G le contexte $g : R\bar{x}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{G, Z \vdash z : \forall x(\forall y(Rx \to Ry) \to \bot)}{G, Z \vdash z : \forall y(R\bar{y} \to Ry) \to \bot}}{G, Z \vdash z : (R\bar{y} \to \forall y Ry) \to R\bar{y}}\,\mathrm{R}}{G, Z \vdash (\mathsf{cc})z : R\bar{y}}}{Z \vdash \lambda g(\mathsf{cc})z : R\bar{x} \to R\bar{y}}}{Z \vdash \lambda g(\mathsf{cc})z : \forall \bar{y}(R\bar{x} \to R\bar{y})} \qquad \dfrac{G, Z \vdash z : \forall x(\forall y(Rx \to Ry) \to \bot)}{G, Z \vdash z : \forall y(R\bar{x} \to Ry) \to \bot}}{\dfrac{Z \vdash (z)\lambda g(\mathsf{cc})z : \bot}{\dfrac{\vdash \lambda z(z)\lambda g(\mathsf{cc})z : \forall x(\forall y(Rx \to Ry) \to \bot) \to \bot}{\vdash \lambda z(z)\lambda g(\mathsf{cc})z : \forall R\{\forall x(\forall y(Rx \to Ry) \to \bot) \to \bot\}}}}$$

ii) Notons Z le contexte $z : \forall x(\forall y(Rx \to Ry) \to \bot)$, K le contexte $k : R\bar{y} \to \bot$ et G le contexte $g : R\bar{x}$.

$$\dfrac{\dfrac{\dfrac{K \vdash k : R\bar{y} \to \bot}{K \vdash k : \forall y(R\bar{y} \to Ry)}\,\mathrm{R} \quad \dfrac{Z \vdash z : \forall x(\forall y(Rx \to Ry) \to \bot)}{Z \vdash z : \forall y(R\bar{y} \to Ry) \to \bot}}{Z, K \vdash (z)k : \bot}}{Z, K \vdash (z)k : \forall y(R\bar{y} \to Ry)} \quad \dfrac{G, Z \vdash z : \forall x(\forall y(Rx \to Ry) \to \bot)}{G, Z \vdash z : \forall y(R\bar{y} \to Ry) \to \bot}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{G, Z, K \vdash (z)(z)k : \bot}{G, Z \vdash \lambda k(z)(z)k : (R\bar{y} \to \bot) \to \bot}}{G, Z \vdash \lambda k(z)(z)k : (R\bar{y} \to \bot) \to R\bar{y}}\,\mathrm{R}}{G, Z \vdash (\mathsf{cc})\lambda k(z)(z)k : R\bar{y}}}{Z \vdash \lambda g(\mathsf{cc})\lambda k(z)(z)k : R\bar{x} \to R\bar{y}}}{Z \vdash \lambda g(\mathsf{cc})\lambda k(z)(z)k : \forall \bar{y}(R\bar{x} \to R\bar{y})}$$

On peut alors conclure comme suit :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{Z \vdash \lambda g(\mathsf{cc})\lambda k(z)(z)k : \forall \bar{y}(R\bar{x} \to R\bar{y})} \quad \dfrac{Z \vdash z : \forall x(\forall y(Rx \to Ry) \to \bot)}{Z \vdash z : \forall y(R\bar{x} \to Ry) \to \bot}}{Z \vdash (z)\lambda g(\mathsf{cc})\lambda k(z)(z)k : \bot}}{\vdash \lambda z(z)\lambda g(\mathsf{cc})\lambda k(z)(z)k : \forall x(\forall y(Rx \to Ry) \to \bot) \to \bot}}{\vdash \lambda z(z)\lambda g(\mathsf{cc})\lambda k(z)(z)k : \forall R\{\forall x(\forall y(Rx \to Ry) \to \bot) \to \bot\}}$$

□

On peut décrire l'exécution du terme associé à la preuve standard de ce théorème :

$$\begin{aligned}
\lambda z(z)\lambda g(\mathsf{cc})z \star \kappa\pi &\succ \kappa \star \lambda g(\mathsf{cc})\kappa \cdot \pi \\
&\succ \lambda g(\mathsf{cc})\kappa \star \kappa_{Ri_1} \cdot \pi_{Rj_1} \\
&\succ \mathsf{cc} \star \kappa \cdot \pi_{Rj_1} \\
&\succ \kappa \star \mathsf{k}_{\pi_{Rj_1}} \cdot \pi_{Rj_1} \\
&\succ \mathsf{k}_{\pi_{Rj}} \star \kappa_{Ri_2} \cdot \pi_{Rj_2} \\
&\succ \kappa_{Ri_2} \star \pi_{Rj_1}
\end{aligned}$$

On voit que la stratégie gagnante de $\exists$ est de jouer $i_2 = j_1$, c'est-à-dire de jouer au deuxième coup ce qu'a joué $\forall$ au premier. Néanmoins, les règles du jeu n'imposent pas à la joueuse de choisir cet individu : celle-ci peut décider de perdre la partie. Les parties associées à cette quasi-preuve sont représentées à la figure 5.1.

Interprétation : La spécification de cette formule peut être interprétée comme décrivant l'échange d'un message entre un client et un serveur avec envoi d'un accusé de réception, échange dans lequel le paquet en question finit par être reçu et acquitté (voir [28]). La quasi-preuve considérée décrit une session où le message est acquitté dès le premier envoi.

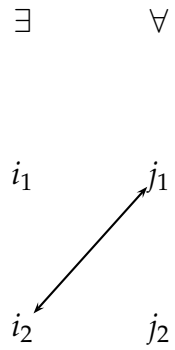


FIG. 5.1 – La partie associée à la quasi-preuve $\lambda z(z)\lambda g(\mathsf{cc})z$.

Observons encore l'exécution du terme associé à la seconde preuve de ce théorème :

$$\begin{aligned}
\lambda z(z)\lambda g(\mathsf{cc})\lambda k(z)(z)k \star \kappa\pi &\succ \kappa \star \lambda g(\mathsf{cc})\lambda k(\kappa)(\kappa)k \cdot \pi \\
&\succ \lambda g(\mathsf{cc})\lambda k(\kappa)(\kappa)k \star \kappa_{Ri_1} \cdot \pi_{Rj_1} \\
&\succ \kappa \star (\kappa)\mathsf{k}_{\pi_{Rj_1}} \cdot \pi_{Rj_1} \\
&\succ (\kappa)\mathsf{k}_{\pi_{Rj_1}} \star \kappa_{Ri_2} \cdot \pi_{Rj_2} \\
&\succ \mathsf{k}_{\pi_{Rj_1}} \star \kappa_{Ri_3} \cdot \pi_{Rj_3} \\
&\succ \kappa_{Ri_3} \star \pi_{Rj_1}
\end{aligned}$$

Dans ce cas, la stratégie gagnante de $\exists$ est de jouer $i_3 = j_1$, c'est-à-dire de jouer au troisième coup ce qu'a joué $\forall$ au premier. Cette quasi-preuve décrit donc une session où le premier exemplaire du message est acquitté, mais seulement après deux autres envois.

5.3.3 La forme de Herbrand du buveur

On considère ici la formule $\exists x(Rx \to R\varphi(x))$. Cette formule est valide puisqu'il suffit de définir x comme étant un individu tel que $\neg R(x)$ s'il en existe un, tout x convenant sinon. Il s'agit de la forme de Herbrand de la formule précédente, telle qu'elle est définie dans la section 7.2. Celle-ci est démontrable en logique intuitionniste.

Théorème 5.3.3.
La quasi-preuve $\lambda z(z)\lambda x(z)\lambda g\ x$ réalise $\forall R\{\forall x((Rx \to R\varphi(x)) \to \bot) \to \bot\}$.

Démonstration. Notons Z le contexte $z : \forall x\{(Rx \to R\varphi(x)) \to \bot\}$.

$$\dfrac{\dfrac{\dfrac{g : Rx_0,\ x : R\varphi(x_0) \vdash x : R\varphi(x_0)}{x : R\varphi(x_0) \vdash \lambda g\ x : Rx_0 \to R\varphi(x_0)} \qquad \dfrac{Z \vdash z : \forall x\{(Rx \to R\varphi(x)) \to \bot\}}{Z \vdash z : (Rx_0 \to R\varphi(x_0)) \to \bot}}{Z, x : R\varphi(x_0) \vdash (z)\lambda g\ x : \bot}}{\dfrac{Z, x : R\varphi(x_0) \vdash (z)\lambda g\ x : R\varphi^2(x_0)}{Z \vdash \lambda x(z)\lambda g\ x : R\varphi(x_0) \to R\varphi^2(x_0)}}$$

Ce qui permet de conclure comme suit :

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{Z \vdash \lambda x(z)\lambda g\ x : R\varphi(x_0) \to R\varphi^2(x_0)} \qquad \dfrac{Z \vdash z : \forall x\{(Rx \to R\varphi(x)) \to \bot\}}{Z \vdash z : (R\varphi(x_0) \to R\varphi^2(x_0)) \to \bot}}{Z \vdash (z)\lambda x(z)\lambda g\ x : \bot}}{\vdash \lambda z(z)\lambda x(z)\lambda g\ x : \forall x\{(Rx \to R\varphi(x)) \to \bot\} \to \bot}}{\vdash \lambda z(z)\lambda x(z)\lambda g\ x : \forall R(\forall x\{(Rx \to R\varphi(x)) \to \bot\} \to \bot)}$$

□

La constante d'interaction associée à $\forall x(Rx \to R\varphi(x)) \to \bot$ sera notée κ_h, elle possède la règle d'exécution suivante :

$$\kappa_h \star \xi \cdot \pi \succ \xi \star \kappa_{Ri} \cdot \pi_{R\varphi(i)}$$

où l'individu i est choisi par la joueuse. Considérant un processus p, celle-ci l'emporte dans la partie associée si l'exécution se termine sur un processus de la forme $\kappa_{Ri} \star \pi_{Ri}$.

On peut décrire l'exécution du terme associé à la preuve standard de ce théorème :

$$\begin{aligned}
\lambda z(z)\lambda x(z)\lambda g\ x \star \kappa_h \cdot \pi &\succ \kappa_h \star \lambda x(\kappa_h)\lambda g x \cdot \pi \\
&\succ \lambda x(\kappa_h)\lambda g\ x \star \kappa_{Ri_1} \cdot \pi_{R\varphi(i_1)} \\
&\succ \kappa_h \star \lambda g\ \kappa_{Ri_1} \cdot \pi_{R\varphi(i_1)} \\
&\succ \lambda g\ \kappa_{Ri_1} \star \kappa_{Ri_2} \cdot \pi_{R\varphi(i_2)} \\
&\succ \kappa_{Ri_1} \star \pi_{R\varphi(i_2)}
\end{aligned}$$

On voit que la stratégie gagnante de $\exists$ est de jouer $i_1 = \varphi(i_2)$; la figure 5.2 représente les parties où elle applique celle-ci.

Le passage d'une formule à sa forme de Herbrand a pour effet sur les jeux associés de faire disparaître $\forall$ de la partie : tout se passe comme si $\exists$ jouait face à une machine, laquelle répond « fonctionnellement » à ses choix. A chaque fois que $\exists$ joue l'individu x, $\forall$ répondra par le choix de $\varphi(x)$. Dans le cas général, le rôle de l'opposant n'est que restreint, puisque celui-ci aura encore à choisir des formules dans l'ensemble $\mathscr{V}$.

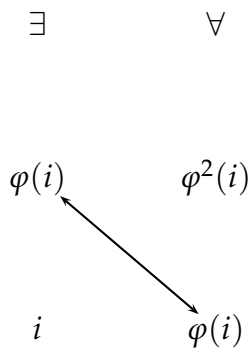


FIG. 5.2 – Partie associée à la quasi-preuve $\lambda z(z)\lambda x(z)\lambda g\ x$.

Interprétation : La session décrite par le terme $\lambda z(z)\lambda g(\mathsf{cc})z$ correspond à l'échange d'un message, mais l'on peut interpréter la connexion établie comme étant défectueuse puisque l'acquittement accepté par le serveur a été envoyé avant les données !

Dans le chapitre 7, nous réaliserons la formule

$$\forall R\exists x(Rx \to R\varphi(x)) \to \forall R\exists x\forall y(Rx \to Ry)$$

qui est un cas particulier du théorème de Herbrand. Nous obtiendrons donc un terme permettant de transformer une stratégie gagnante face à un adversaire se comportant de manière fonctionnelle, en une stratégie gagnante face à un adversaire quelconque.

Chapitre 6

L'axiome du choix au premier ordre

6.1 Enoncé et mode opératoire

L'axiome du choix dépendant a été réalisé dans [25]. On trouvera dans [16] comment réaliser l'axiome du choix sous sa forme la plus générale.
Ce chapitre se réfère à [17], où est réalisé un axiome dit *du choix non-extensionnel au second ordre*. On montre ici que l'instruction utilisée pour réaliser cet axiome permet également de réaliser une formulation de l'axiome du choix au niveau des individus d'un modèle. Celle-ci sera utilisée dans le chapitre 7 pour réaliser un cas particulier du théorème de Herbrand.

On considère le schéma d'axiome suivant, qui exprime l'axiome du choix sur les individus du modèle considéré :

Pour toute formule $F(x,y)$ à deux variables libres x et y, il existe une application ϕ telle que

$$\forall x \left[\exists y F(x,y) \rightarrow F(x,\phi(x))\right].$$

Celui-ci est évidemment conséquence de l'axiome de récurrence, puisqu'on peut alors définir $\phi(x)$ comme étant le plus petit entier z tel que $F(x,z)$ si un tel entier existe, n'importe quel entier convenant sinon. Mais l'axiome de récurrence n'est pas satisfait dans les modèles de la réalisabilité.

Ce schéma implique le choix dépendant sur les individus. Il suffit en effet d'itérer l'application ϕ :

Si la formule $\forall x \exists y F(x,y)$ est vraie, alors il existe une application ϕ telle que pour tout individu x et tout entier n, la formule $F(\phi^n(x),\phi^{n+1}(x))$ est encore vraie.

Néanmoins, le schéma considéré n'implique pas l'existence d'un bon ordre sur les individus.

Nous allons dans ce chapitre réaliser une formule impliquant ce schéma

d'axiome. Il faudra pour cela considérer de nouvelles instructions.

On considère un ensemble $\mathscr{N}$ isomorphe au modèle standard de l'arithmétique, qui est l'ensemble des termes donnés par la grammaire suivante :

$$\tau := a | \varphi(\tau)$$

Définition 6.1.1.
Pour tout $x \in \mathscr{N}$, on notera $\underline{x}$ l'entier de Church associé à l'unique entier n tel que $x = \varphi^n(a)$.

Il est clair que si $x \in \mathscr{N}$, l'entier de Church $\underline{x}$ réalise la formule suivante, notée $Ent(x)$:

$$\forall X \{\forall y (Xy \to X\varphi(y)), Xa \to Xx\}.$$

Si $\mathscr{M}$ est alors un modèle de la réalisabilité, on appellera *entier du modèle* tout élément d'un tel modèle pour lequel la formule Ent est vérifiée.

Définition 6.1.2.
On note $\lhd$ l'ordre total défini sur $\mathscr{N}$ par

$$x \lhd y \text{ ssi } \exists m \in \mathbb{N}^*, \varphi^m(x) = y.$$

On fixe ensuite une bijection :

$$\begin{array}{rcl} \mathscr{N} & \to & \Pi \\ x & \mapsto & \pi_x \end{array}$$

et l'on considère une instruction σ possédant la règle d'exécution suivante :

$$\sigma \star t \cdot \pi \succ t \star \underline{x} \cdot \pi \text{ où } x \in \mathscr{N} \text{ est tel que } \pi = \pi_x$$

Cette instruction permet de réaliser l'expression au second ordre du schéma considéré, pour les ensembles $\perp\!\!\!\perp$ saturés pour celle-ci.

Lemme 6.1.3.
Soit F une formule avec deux variables libres, et $\perp\!\!\!\perp$ un ensemble saturé pour σ. Il existe une application $f : \mathscr{N}^2 \to \mathscr{N}$ telle que σ réalise

$$\forall x \left(\forall n \left[Ent(n) \to F(x, f(x,n)) \right] \to \forall y F(x,y) \right)$$

Démonstration. On considère d'abord une application f possédant la propriété suivante

$$\forall x \in \mathscr{N}, \forall z \in \mathscr{N}, \pi_z \in ||\forall y F(x,y)|| \implies \pi_z \in ||F(x, f(x,z))||.$$

Une telle application se construit par une application directe de l'axiome du choix dénombrable dans $\mathscr{N}$. En effet, on a $||\forall y F(x,y)|| = \bigcup_{y \in \mathscr{N}} ||F(x,y)||$, et l'on

peut donc définir $f(x,z)$ comme étant un individu y tel que $\pi_z \in ||F(x,y)||$ si un tel individu existe, n'importe quel individu sinon.
On fixe alors $x \in \mathscr{N}$, un terme t réalisant $\forall n[Ent(n) \rightarrow F(x,f(x,n))]$ ainsi qu'une pile $\pi = \pi_z$ dans $||\forall y F(x,y)||$. On doit montrer que le processus $\sigma \star t \cdot \pi_z$ est dans $\perp\!\!\!\perp$. Mais celui-ci se réduit en $t \star \underline{z} \cdot \pi_z$, avec π_z dans $||F(x,f(x,z))||$ par définition de f, et ce dernier processus est dans $\perp\!\!\!\perp$ puisque t réalise $Ent(z) \rightarrow F(x,f(x,z))$. Considérant que $\perp\!\!\!\perp$ est saturé pour σ, on en déduit le résultat. □

L'axiome du choix dépendant au premier ordre se déduit en logique classique de la formule $\forall x \left(\forall n \left[Ent(n) \rightarrow F(x,f(x,n)) \right] \rightarrow \forall y F(x,y) \right)$: il suffit de définir $\phi(x)$ comme étant $f(x,n)$, pour n le plus petit des entiers l du modèle tel que $\neg F(x,f(x,l))$ s'il en existe un, et 0 sinon. On obtient alors le schéma suivant :

$$\forall x(F(x,\phi(x)) \rightarrow \forall y F(x,y)),$$

ce qui donne le schéma considéré après contraposition. Néanmoins, l'application ϕ ne sera plus représentée par un symbole de notre langage mais par un prédicat binaire fonctionnel.

On peut donc, en utilisant le lemme d'adéquation, écrire une quasi-preuve contenant σ qui réalisera cet axiome, et considérer que l'instruction σ représente le contenu opérationnel de celui-ci. Nous allons détailler cette construction dans la partie suivante, afin d'obtenir un terme le plus simple possible.

Interprétation : On peut interpréter l'instruction σ comme étant un algorithme de signature. Elle permet en effet d'associer à chaque pile un entier de manière univoque, sans que l'application réciproque soit utilisable.

Remarque : On pourrait également utiliser une instruction σ' numérotant les termes en lieu et place des piles, mais cela donnerait *in fine* une quasi-preuve plus complexe.

6.2 Formulation et réalisation du schéma

Quatre nouvelles instructions

On considère quatre instructions notées γ, E, U_0 et U_1 définies par les règles d'exécution suivantes :

$$\begin{array}{rcl}
\gamma \star t \cdot \pi & \succ & E \star t \cdot \underline{n} \cdot \pi \text{ où } n \text{ est le numéro de la pile } \pi \\
E \star t \cdot u \cdot \pi & \succ & t \star ((U_0)(E)t)u\mathsf{k}_\pi \cdot ((U_1)(E)t)u\mathsf{k}_\pi \cdot \pi \\
U_0 \star t \cdot u \cdot v \cdot w \cdot \pi & \succ & w \star t \cdot u \cdot v \cdot \pi \\
U_1 \star \eta \cdot \underline{n} \cdot \mathsf{k}_\pi \cdot u \cdot \eta' \cdot \underline{n'} \cdot \mathsf{k}_{\pi'} \cdot \rho & \succ & \begin{cases} \eta' \star \underline{n} \cdot \pi & \text{si } n \lhd n' \\ \eta \star \underline{n'} \cdot \pi' & \text{si } n \rhd n' \\ u \star \rho & \text{sinon} \end{cases}
\end{array}$$

On pourrait aisément écrire avec les seules instructions cc et σ quatre quasi-preuves possédant ces propriétés de réduction ; on utilise néanmoins des instructions supplémentaires afin de faciliter la compréhension du terme réalisant le schéma d'axiome considéré.

Le schéma et sa réalisation

On peut évidemment rajouter des paramètres à la formule F considérée. Après contraposition, le schéma d'axiome considéré s'exprime donc comme suit :

Pour toute formule $F(\overrightarrow{n}, \overrightarrow{\Psi}, x, y)$ à deux variables libres x et y, il existe une application ϕ telle que

$$\forall \overrightarrow{n} \forall \overrightarrow{R} \forall x \left[F(\overrightarrow{n}, \overrightarrow{R}, x, \phi(\overrightarrow{n}, \overrightarrow{R}, x)) \to \forall y F(\overrightarrow{n}, \overrightarrow{R}, x, y) \right].$$

Ce qui donne en logique du second ordre l'énoncé suivant, où les paramètres ont été effacés pour plus de clarté :

Pour toute formule $F(x, y)$ à deux variables libres x et y, on a

$$\exists Z \forall x \left[\exists z \left(\forall y \{ Z(x,y) \to y = z \}, \forall y \{ y = z \to Z(x,y) \} \right) \right.$$
$$\left. \bigwedge \forall z \left(F(x,z), Z(x,z) \to \forall y F(x,y) \right) \right].$$

On peut en fait construire le prédicat noté Z ci-dessus en fonction de la formule F, ce qui permet de réaliser par une quasi-preuve *indépendante de la formule F choisie* la formule suivante :

$$\forall x \left[\exists z \left(\forall y\{Z(x,y) \to y = z\}, \forall y\{y = z \to Z(x,y)\} \right) \right.$$
$$\left. \bigwedge \forall z \left(F(x,z), Z(x,z) \to \forall y F(x,y) \right) \right].$$

Pour simplifier la quasi-preuve obtenue, nous allons dans cette partie démontrer que γ réalise le schéma équivalent suivant :

Pour toute formule $F(x,y)$ à deux variables libres, il existe un prédicat binaire V tel que

$$\forall x \left[\forall z \left(\forall y\{V(x,y) \to y = z\}, \forall y\{y = z \to V(x,y)\} \to F(x,z) \right) \to \forall z F(x,z) \right].$$

Cet énoncé exprime que pour tout individu x, si la formule $\forall y F(x,y)$ est fausse, alors $V(x,.)$ est un singleton, et l'unique élément z qu'il contient est tel que $\neg F(x,z)$ est vérifiée. On peut donc en déduire la formulation précédente, en définissant Z par $Z(x,y) \equiv V(x,y)$ si $\forall y F(x,y)$ est fausse, et $Z(x,y) \equiv y = a$ sinon.

Définitions/Notations : Considérons jusqu'à la fin de cette partie une formule à paramètres $F(x,y)$, dont x et y sont les seules variables libres.
On fixe d'abord une application $f : \mathcal{N}^2 \to \mathcal{N}$ possédant la propriété suivante

$$\forall x \in \mathcal{N}, \forall z \in \mathcal{N}, \pi_z \in ||\forall y F(x,y)|| \implies \pi_z \in ||F(x, f(x,z))||,$$

construite en utilisant l'axiome du choix dépendant sur $\mathcal{N}$.
On appelle alors V le prédicat binaire défini, quels que soient les éléments x et y de $\mathcal{N}$, par

$$V(x,y) = \left|\left| \forall n \left(\bigcap_{m \lhd n} |\{\underline{m}\} \to F(x, f(x,m))|, \{\underline{n}\}, \Phi(n,x) \to y = f(x,n) \right) \right|\right|$$

où $\Phi(n,x) = \{\mathsf{k}_\pi ; \pi \in ||F(x, f(x,n))||\}$.

Le sens de ce prédicat est le suivant : s'il existe un plus petit entier n du modèle tel que $\neg F(x, f(x,n))$, V associe à x l'élément $f(x,n)$, et sinon V associe tous les individus de la forme $f(x,p)$ à x. C'est-à-dire qu'aucun choix n'est fait lorsque cela n'est pas nécessaire : c'est ce « minimalisme » dans la définition de V qui permet de simplifer la quasi-preuve obtenue γ en modifiant la formulation de l'axiome.

Remarque : On se reportera à [18], où est réalisée l'équivalence entre chaque formule de la forme $\neg F \to G$ et la formule $\{\mathsf{k}_\pi ; \pi \in ||F||\} \to G$.

Avec ces définitions, on va pouvoir démontrer le résultat annoncé. Il faut pour cela se restreindre aux ensembles $\perp\!\!\!\perp$ saturés pour chacune des quatre nouvelles instructions considérées.

Théorème 6.2.1.
L'instruction γ réalise la formule

$$\forall x \left[\forall z \left(\forall y \{V(x,y) \to y = z\}, \forall y \{y = z \to V(x,y)\} \to F(x,z) \right) \to \forall z F(x,z) \right].$$

La présence éventuelle de paramètres $\overrightarrow{n}$ et $\overrightarrow{R}$ dans la formule F ne change rien à la démonstration, seul le prédicat V en dépend. On montre d'abord le résultat suivant.

Lemme 6.2.2.
L'instruction E réalise la formule

$$\forall x \forall n [H(x), \{\underline{n}\} \to F(x, f(x,n))]$$

où $H(x)$ désigne la formule suivante :

$$\forall p \left(\forall y \{V(x,y) \to y = f(x,p)\}, \forall y \{y = f(x,p) \to V(x,y)\} \to F(x, f(x,p)) \right).$$

Démonstration. Soit $x \in \mathscr{N}$. On va démontrer par induction (relativement à $\lhd$) que quel que soit l'élément n de $\mathscr{N}$, le terme t réalisant $H(x)$ et la pile π dans $||F(x, f(x,n))||$, le processus $E \star t \cdot \underline{n} \cdot \pi$ est dans $\perp\!\!\!\perp$.
Ce processus se réduisant en $t \star ((U_0)(E)t)\underline{n}\mathsf{k}_\pi \cdot ((U_1)(E)t)\underline{n}\mathsf{k}_\pi \cdot \pi$, il nous suffit de montrer considérant que $\perp\!\!\!\perp$ est saturé pour l'instruction E que :
i) $((U_0)(E)t)\underline{n}\mathsf{k}_\pi$ réalise $\forall y \{V(x,y) \to y = f(x,n)\}$.
ii) $((U_1)(E)t)\underline{n}\mathsf{k}_\pi$ réalise $\forall y \{y = f(x,n) \to V(x,y)\}$.

Pour démontrer *i)*, on considère $y \in \mathscr{N}$ ainsi qu'un terme u réalisant $V(x,y)$ et une pile ρ dans $||y = f(x,n)||$. Le processus $((U_0)(E)t)\underline{n}\mathsf{k}_\pi \star u \cdot \rho$ se réduit en $u \star (E)t \cdot \underline{n} \cdot \mathsf{k}_\pi \cdot \rho$, ce qui donne le résultat considérant que $\perp\!\!\!\perp$ est saturé pour U_0, que $u \Vdash \bigcap_{m \lhd n} |\{\underline{m}\} \to F(x, f(x,m))|, \{\underline{n}\}, \Phi(n,x) \to y = f(x,n)$ et que par hypothèse d'induction $(E)t$ réalise $\{\underline{m}\} \to F(x, f(x,m))$ pour tout $m \lhd n$.

Pour *ii)*, on considère $y \in \mathscr{N}$, un terme u réalisant $y = f(x,n)$ ainsi qu'une pile ρ' dans $V(x,y)$. Il existe donc $n' \in \mathscr{N}$, $t' \in \bigcap_{m \lhd n'} |\{\underline{m}\} \to F(x, f(x,m))|$, $\pi' \in ||F(x, f(x,n'))||$ et $\rho \in ||y = f(x,n')||$ tels que $\rho' = t' \cdot \underline{n}' \cdot \mathsf{k}_{\pi'} \cdot \rho$. Il nous faut alors montrer $((U_1)(E)t)\underline{n}\mathsf{k}_\pi \star u \cdot t' \cdot \underline{n}' \cdot \mathsf{k}_{\pi'} \cdot \rho \in \perp\!\!\!\perp$. Mais ce processus se réduit

en $U_1 \star (E)t \cdot \underline{n} \cdot \mathsf{k}_\pi \cdot u \cdot t' \cdot \underline{n}' \cdot \mathsf{k}_{\pi'} \cdot \rho$, et la suite de l'exécution dépend de l'ordre relatif entre n et n'. Dans chacun des cas on obtient le résultat considérant que $\perp\!\!\!\perp$ est saturé pour U_1.
Si $n = n'$, ce processus se réduit en $u \star \rho$, qui est dans $\perp\!\!\!\perp$ car u réalise alors $y = f(x, n')$.
Si $n \lhd n'$, ce processus se réduit en $t' \star \underline{n} \cdot \pi$ qui est dans $\perp\!\!\!\perp$ car t' est dans $\bigcap_{m \lhd n'} |\{\underline{m}\} \to F(x, f(x, m))|$, et réalise donc $\{\underline{n}\} \to F(x, f(x, n))$.
Si $n' \lhd n$, ce processus se réduit en $E \star t \cdot \underline{n'} \cdot \pi'$ qui est dans $\perp\!\!\!\perp$, par hypothèse d'induction.

□

On peut alors conclure.

Démonstration du théorème 6.2.1. Soit $x \in \mathscr{N}$. On considère un terme t réalisant

$$\forall z \left(\forall y \{V(x,y) \to y = z\}, \forall y \{y = z \to V(x,y)\} \to F(x,z) \right)$$

ainsi qu'une pile $\pi = \pi_n$ dans $||\forall z F(x,z)||$. On doit montrer que le processus $\gamma \star t \cdot \pi$ est dans $\perp\!\!\!\perp$. Celui-ci se réduisant en $E \star t \cdot \underline{n} \cdot \pi$, il suffit considérant que $\perp\!\!\!\perp$ est saturé pour γ de montrer $E \star t \cdot \underline{n} \cdot \pi \in \perp\!\!\!\perp$. Mais comme $\pi = \pi_n$, on a $\pi \in ||F(x, f(x,n))||$, et comme par ailleurs il est clair que t réalise

$$\forall p \left(\forall y \{V(x,y) \to y = f(x,p)\}, \forall y \{y = f(x,p) \to V(x,y)\} \to F(x, f(x,p)) \right)$$

le lemme précédent permet de conclure que le processus $E \star t \cdot \underline{n} \cdot \pi$ est dans $\perp\!\!\!\perp$, ce qui donne le résultat. □

Signalons enfin que les seules opérations qui seront effectuées sur les entiers produits par l'instruction γ sont des comparaisons, via l'instruction U_1. Il est évidemment hors de propos de calculer le successeur d'un tel entier. On pourrait en fait réaliser l'axiome du choix au premier ordre en utilisant une seule instruction, notée U, dont la règle d'exécution est la suivante :

$$U \star \eta \cdot \mathsf{k}_{\pi_n} \cdot u \cdot \eta' \cdot \mathsf{k}_{\pi_{n'}} \cdot \rho \succ \begin{cases} \eta' \star \pi_n & \text{si } n \lhd n' \\ \eta \star \pi_{n'} & \text{si } n \rhd n' \\ u \star \rho & \text{sinon} \end{cases}$$

L'instruction présentée ici a néanmoins été choisie car elle permet une démarche plus pédagogique.

Chapitre 7

Le théorème de Herbrand

Les principaux résultats de cette section se trouvent dans [10].

7.1 Enoncé usuel du théorème

Le théorème de Herbrand établit que pour toute formule du premier ordre F, il existe une formule existentielle $F^{\mathscr{H}}$ telle que :

i) La formule $F \to F^{\mathscr{H}}$ est démontrable.

ii) Si la formule $F^{\mathscr{H}}$ est démontrable, alors la formule F l'est également.

Mais la plupart des applications de ce théorème n'utilisent qu'un énoncé restreint. Nous allons dans la suite considérer le cas typique suivant :

Etant donné un langage $\mathscr{L}$ et une formule prénexe du premier ordre de la forme $\exists x \forall y F(x,y)$ écrite dans $\mathscr{L}$, si φ désigne un symbole de fonction n'appartenant pas à $\mathscr{L}$, alors :

i) *La formule $\exists x \forall y F(x,y) \to \exists x F(x, \varphi x)$ est démontrable.*

ii) *Si la formule $\exists x F(x, \varphi x)$ est démontrable, alors la formule $\exists x \forall y F(x,y)$ l'est également.*

La formule $\exists x F(x, \varphi x)$ est appelée la *forme de Herbrand* de $\exists x \forall y F(x,y)$. La forme de Herbrand d'une formule G est équivalente à la négation de la forme de Skolem de $\neg G$.

Dans ce chapitre, nous allons réaliser une formule exprimant ce cas particulier du théorème de Herbrand. Cela donnera un premier exemple d'utilisation de l'instruction γ associée à l'axiome du choix au premier ordre. Cette formulation aura également l'intérêt de posséder une interprétation dans le cadre des protocoles réseaux.

7.2 Forme de Herbrand, forme de Skolem

On considère dans la suite de ce chapitre les formules normales données dans la définition 5.1.1.

Dans cette section, on définit inductivement la forme de Herbrand d'une formule normale ; cette définition se fait simultanément à celle de sa forme de Skolem. Elle donne des formules légèrement plus simples que celles définies à partir des formes prénexes.

La forme de Herbrand (resp. de Skolem) d'une formule normale Φ est obtenue en effaçant ses quantificateurs universels positifs (resp. négatifs), et en remplaçant les variables qu'ils lient par des termes dépendant des variables libres de Φ.

Notation : La formule $\forall x(Fx \to \bot) \to \bot$ sera noté $\exists^* x\ Fx$.

Définition 7.2.1.
Soit $F = \forall x_1 \dots \forall x_k(\Phi_1[x_1, \dots, x_k], \dots, \Phi_p[x_1, \dots, x_k] \to A[x_1, \dots, x_k])$ *une formule normale. Soient* $y_1, \dots, y_n$ *ses variables libres. On définit récursivement les formules (normales) notées* $F^{\mathscr{H}}$ *et* $F^{\mathscr{S}}$*, appelées respectivement « forme de Herbrand de F » et « forme de Skolem de F », par les règles suivantes :*

i) $F^{\mathscr{S}} = \forall x_1 \dots \forall x_k({\Phi_1}^{\mathscr{H}}[x_1, \dots, x_k], \dots, {\Phi_p}^{\mathscr{H}}[x_1, \dots, x_k] \to A[x_1, \dots, x_k])$.

ii) $F^{\mathscr{H}} = {\Phi_1}^{\mathscr{S}}[\tau_1, \dots, \tau_k], \dots, {\Phi_p}^{\mathscr{S}}[\tau_1, \dots, \tau_k] \to A[\tau_1, \dots, \tau_k]$*, où on a posé* $\tau_j = f_j(y_1, \dots, y_n)$*, les* f_j *étant des symboles de fonctions d'arité n deux-à-deux distincts, qui sont de plus supposés ne pas apparaître dans les formules* $\Phi_1^{\mathscr{S}}, \dots, \Phi_p^{\mathscr{S}}, A$.

Remarques : Les formes de Herbrand et de Skolem d'une formule ne sont pas définies de manière univoque, puisqu'une liberté est laissée dans le choix des symboles de fonctions employés. Par abus de langage, on parlera néanmoins de *la* forme de Herbrand ou de Skolem d'une formule.
Cette définition est celle qui sera utilisée ici. Bien évidemment, si l'on se place dans le cadre de la logique mathématique usuelle, on considère que les symboles de fonctions f_j employés n'appartiennent pas au langage considéré.

Exemples : Si $F = \forall x \exists^* y \forall z G(x, y, z)$ est une formule close écrite sous forme prénexe, alors $F^{\mathscr{S}} = \forall x \neg\neg \forall z G(x, f(x), z)$ où f est un symbole de fonction unaire, et $F^{\mathscr{H}} = \exists^* y G(g_0, y, g_1(y))$, où g_0 et g_1 désignent des symboles de fonctions d'arité respective 0 et 1. Les formes de Herbrand et de Skolem s'accordent dans ce cas avec leurs définitions usuelles.
Mais si l'on considère la formule valide $F = \exists^* x \forall y\{(G \to Px) \to Py\}$, où $G = \exists^* x' \forall y'\{Rx' \to Ry'\}$, on vérifie que

$F^{\mathscr{H}} = \exists^* x\{(\exists^* x'[Rx' \to Rg(x')] \to Px) \to Pf(x)\}$. Cependant, la forme prénexe de F est $\exists^* x \forall y \exists^* x' \forall y' \{([Rx' \to Ry'] \to Px) \to Py\}$, et sa forme de Herbrand est usuellement définie comme étant la formule
$\exists^* x \exists^* x' \{([Rx' \to Rg(x,x')] \to Px) \to Pf(x)\}$. La définition donnée ici permet dans ce cas d'utiliser deux symboles de fonctions unaires, en lieu et place d'un symbole unaire et d'un symbole binaire.

On dispose du résultat suivant, qui assure l'adéquation entre cette définition des formes de Herbrand/Skolem et leur définition usuelle.

Définition 7.2.2.
On appelle formule existentielle toute formule qui s'écrit $\exists^* x_1 \dots \exists^* x_n F(x_1, \dots, x_n)$ où F est une formule écrite sans quantificateur du premier ordre.
De même, on appelle formule universelle toute formule qui s'écrit $\forall x_1 \dots \forall x_n F(x_1, \dots, x_n)$ où F est une formule écrite sans quantificateur du premier ordre.

Théorème 7.2.3.
Soit F une formule normale. On peut prouver en logique classique que $F^{\mathscr{S}}$ est équivalente à une formule universelle, et que $F^{\mathscr{H}}$ est équivalente à une formule existentielle.

Démonstration. On démontre simultanément les deux résultats par induction sur F.
Si F est atomique, on a $F = F^{\mathscr{H}} = F^{\mathscr{S}}$, ce qui prouve le résultat.
Si F s'écrit $\forall x_1 \dots \forall x_k(\Phi_1[x_1, \dots, x_k], \dots, \Phi_p[x_1, \dots, x_k] \to A[x_1, \dots, x_k])$, on a par définition que $F^{\mathscr{H}}$ s'écrit de la manière suivante :

$$\Phi_1{}^{\mathscr{S}}[\tau_1, \dots, \tau_k], \dots, \Phi_p{}^{\mathscr{S}}[\tau_1, \dots, \tau_k] \to A[\tau_1, \dots, \tau_k].$$

L'hypothèse d'induction assure que chacune des formules $\Phi_i{}^{\mathscr{S}}[\tau_1, \dots, \tau_k]$ est équivalente à une formule de la forme $\forall z_i^1 \dots \forall z_i^{m_i} \Psi_i$, que l'on notera $\forall \overrightarrow{z_i} \Psi_i$. $F^{\mathscr{H}}$ est alors équivalente à $\forall \overrightarrow{z_1} \Psi_1, \dots, \forall \overrightarrow{z_p} \Psi_p \to A[\tau_1, \dots, \tau_k]$. Il reste alors à démontrer, considérant que chacun des quantificateurs universels apparaissant dans cette formule apparaissent en position négative, qu'une telle formule est équivalente à

$$\exists^* \overrightarrow{z_1} \dots \exists^* \overrightarrow{z_p} (\Psi_1, \dots, \Psi_p \to A[\tau_1, \dots, \tau_k])$$

ce qui se fait aisément par induction sur p.
On démontrerait de même que $F^{\mathscr{S}}$ est équivalente à une formule universelle. □

Exemples : Si $F = \exists^* x \forall y \{(G \to Px) \to Py\}$, où $G = \exists^* x' \forall y' \{Rx' \to Ry'\}$, on vérifie que $F^{\mathscr{H}} = \exists^* x\{(\exists^* x'[Rx' \to Rg(x')] \to Px) \to Pf(x)\}$ qui est équivalente à $\exists^* x \exists^* x' \{([Rx' \to Rg(x')] \to Px) \to Pf(x)\}$.

Mais le résultat important est le suivant.

Théorème 7.2.4.
Soit F une formule normale close. Alors $||F|| \subseteq ||F^{\mathscr{S}}||$ et $||F|| \supseteq ||F^{\mathscr{H}}||$.

Le résultat découle du lemme suivant, adapté à une preuve par induction.

Lemme 7.2.5.
Soit F une formule normale, dont les variables libres sont parmi $y_1, \ldots, y_n$. Soient $\tau_1, \ldots, \tau_n$ des termes clos. Alors $||F[\tau_1/y_1, \ldots, \tau_n/y_n]|| \subseteq ||F^{\mathscr{S}}[\tau_1/y_1, \ldots, \tau_n/y_n]||$ et $||F[\tau_1/y_1, \ldots, \tau_n/y_n]|| \supseteq ||F^{\mathscr{H}}[\tau_1/y_1, \ldots, \tau_n/y_n]||$.

Démonstration. Si F est atomique, le résultat est trivial.
Supposons maintenant $F = \forall x_1 \ldots \forall x_k(\Phi_1, \ldots, \Phi_p \rightarrow A)$. Sa forme de Herbrand s'écrit

$$\Phi_1{}^{\mathscr{S}}[\sigma_1/x_1, \ldots, \sigma_k/x_k], \ldots, \Phi_p{}^{\mathscr{S}}[\sigma_1/x_1, \ldots, \sigma_k/x_k] \rightarrow A[\sigma_1/x_1, \ldots, \sigma_k/x_k],$$

où $\sigma_1, \ldots, \sigma_k$ sont des termes dont les variables libres sont parmi $y_1, \ldots, y_n$. On notera σ'_i le terme clos $\sigma_i[\tau_1/y_1, \ldots, \tau_n/y_n]$.
L'ensemble $||F^{\mathscr{H}}[\tau_1/y_1, \ldots, \tau_n/y_n]||$ est alors égal à

$$||\Phi_1{}^{\mathscr{S}}[\sigma'_1/x_1, \ldots, \sigma'_k/x_k, \tau_1/y_1, \ldots, \tau_n/y_n], \ldots, \Phi_p{}^{\mathscr{S}}[\sigma'_1/x_1, \ldots, \sigma'_k/x_k, \tau_1/y_1, \ldots, \tau_n/y_n] \rightarrow A[\sigma'_1/x_1, \ldots, \sigma'_k/x_k, \tau_1/y_1, \ldots, \tau_n/y_n]||.$$

Mais l'hypothèse d'induction assure

$$||\Phi_i{}^{\mathscr{S}}[\sigma'_1/x_1, \ldots, \sigma'_k/x_k, \tau_1/y_1, \ldots, \tau_n/y_n]|| \supseteq ||\Phi_i[\sigma'_1/x_1, \ldots, \sigma'_k/x_k, \tau_1/y_1, \ldots, \tau_n/y_n]||.$$

La flèche étant contravariante à gauche (voir le lemme 2.1.24) on obtient que $||F^{\mathscr{H}}[\tau_1/y_1, \ldots, \tau_n/y_n]||$ est inclus dans

$$||\Phi_1[\sigma'_1/x_1, \ldots, \sigma'_k/x_k, \tau_1/y_1, \ldots, \tau_n/y_n], \ldots, \Phi_p[\sigma'_1/x_1, \ldots, \sigma'_k/x_k, \tau_1/y_1, \ldots, \tau_n/y_n] \rightarrow A[\sigma'_1/x_1, \ldots, \sigma'_k/x_k, \tau_1/y_1, \ldots, \tau_n/y_n]||,$$

qui est lui-même inclus dans

$$\Big|\Big|\forall x_1 \ldots \forall x_k \Big(\Phi_1[x_1, \ldots, x_k, \tau_1/y_1, \ldots, \tau_n/y_n], \ldots, \Phi_n[x_1, \ldots, x_k, \tau_1/y_1, \ldots, \tau_n/y_n] \rightarrow A[x_1, \ldots, x_k, \tau_1/y_1, \ldots, \tau_n/y_n]\Big)\Big|\Big|.$$

Ce dernier ensemble étant égal à $||F[\tau_1/y_1, \ldots, \tau_n/y_n]||$, on obtient le résultat dans ce cas.

Un raisonnement similaire permet de démontrer la seconde affirmation. En effet, l'ensemble $||F^{\mathscr{S}}[\tau_1/y_1,\ldots,\tau_n/y_n]||$ est égal à

$$\Big|\Big|\Big\{\forall x_1\ldots\forall x_k\ \Big(\Phi_1{}^{\mathscr{H}}[x_1,\ldots,x_k],\ldots,\Phi_n{}^{\mathscr{H}}[x_1,\ldots,x_k] \rightarrow A[x_1,\ldots,x_k]\Big)\Big\}[\tau_1/y_1,\ldots,\tau_n/y_n]\Big|\Big|,$$

soit encore à

$$\Big|\Big|\forall x_1\ldots\forall x_k\ \Big(\Phi_1{}^{\mathscr{H}}[x_1,\ldots,x_k,\tau_1/y_1,\ldots,\tau_n/y_n],\ldots,\Phi_n{}^{\mathscr{H}}[x_1,\ldots,x_k,\tau_1/y_1,\ldots,\tau_n/y_n] \rightarrow A[x_1,\ldots,x_k,\tau_1/y_1,\ldots,\tau_n/y_n]\Big)\Big|\Big|,$$

qui est par définition égal à

$$\bigcup_{p_1,\ldots,p_k\in\mathscr{N}} ||\Phi_1{}^{\mathscr{H}}[p_1,\ldots,p_k,\tau_1/y_1,\ldots,\tau_n/y_n],\ldots,\Phi_n{}^{\mathscr{H}}[p_1,\ldots,p_k,\tau_1/y_1,\ldots,\tau_n/y_n] \rightarrow A[p_1,\ldots,p_k,\tau_1/y_1,\ldots,\tau_n/y_n]||.$$

L'hypothèse d'induction assure alors que quels que soient les éléments $p_1,\ldots,p_k$ de $\mathscr{N}$ on a

$$||\Phi_1{}^{\mathscr{H}}[p_1,\ldots,p_k,\tau_1/y_1,\ldots,\tau_n/y_n]|| \subseteq ||\Phi_1[p_1,\ldots,p_k,\tau_1/y_1,\ldots,\tau_n/y_n]||,$$

ce qui étant donné la contravariance de la flèche à gauche montre que l'ensemble considéré contient

$$\bigcup_{p_1,\ldots,p_k\in\mathscr{N}} ||\Phi_1[p_1,\ldots,p_k,\tau_1/y_1,\ldots,\tau_n/y_n],\ldots,\Phi_n[p_1,\ldots,p_k,\tau_1/y_1,\ldots,\tau_n/y_n] \rightarrow A[p_1,\ldots,p_k,\tau_1/y_1,\ldots,\tau_n/y_n]||.$$

Ce dernier ensemble étant égal à $||F[\tau_1/y_1,\ldots,\tau_n/y_n]||$, on a démontré le résultat. □

Corollaire 7.2.6.
Quelle que soit la formule normale close F, $I \Vdash F^{\mathscr{S}} \rightarrow F$ et $I \Vdash F \rightarrow F^{\mathscr{H}}$.

7.3 Le théorème de Herbrand dans un cas particulier

7.3.1 Formulation et preuve du théorème

On considère dans la suite une formule $F(U,V)$, possédant les variables propositionnelles U et V comme seules variables libres. On considère les deux formules suivantes, qui expriment un cas particulier du théorème de Herbrand :
i) $\forall R\{\exists^* x \forall y F(Rx, Ry) \to \exists^* x F(Rx, R\varphi(x))\}$.
ii) $\forall R \exists^* x F(Rx, R\varphi(x)) \to \forall R \exists^* x \forall y F(Rx, Ry)$.

Ce théorème permet de donner un premier exemple d'utilisation de l'instruction associée à l'axiome du choix dépendant, qui pourra être interprété en termes de protocoles réseaux.

Dans cette section, on appelle $\mathscr{N}$ l'ensemble des termes donnés par un symbole de constante a et un symbole de fonction φ, c'est-à-dire les termes donnés par la grammaire suivante :

$$\tau := a | \varphi(\tau)$$

On développe alors la théorie de la réalisabilité classique à partir de cet ensemble. On dénotera toujours par $\lhd$ la relation d'ordre totale sur $\mathscr{N}$ définie par :

$$x \lhd y \text{ ssi } \exists m \in \mathbb{N}^*, \varphi^m(x) = y.$$

On dénotera dans ce qui suit par $x \simeq a$ la formule $\forall z(\varphi(z) \neq x)$.

On remarque déjà que le corollaire 7.2.6 assure que la formule *i)* est réalisée par $\lambda x\, x$. Au contraire, la formule écrite en *ii)* n'est pas réalisée par une quasi-preuve indépendante de la formule F considérée, puisque la preuve dépend de la structure de cette formule.

Considérons la formule suivante, notée $ext(F)$ (pour *extensionnalité de* F) :

$$\forall U \forall V \forall U' \forall V' \Big((U \to U'), (U' \to U), (V \to V'), (V' \to V), F(U,V) \to F(U',V') \Big)$$

qui exprime que $F(U,V)$ ne dépend que des variables propositionnelles U et V, c'est-à-dire que F est un prédicat extensionnel du troisième ordre. Pour toute formule F, $ext(F)$ est démontrable en logique classique du second ordre (il suffit de raisonner par induction sur F).

Nous allons dans la suite montrer que la formule suivante est réalisée par une quasi-preuve, indépendamment de la formule F :

$$ext(F), \forall R \exists^* x F(Rx, R\varphi(x)) \to \forall R \exists^* x \forall y F(Rx, Ry)$$

Indiquons d'abord comment démontrer ce résultat en logique classique grâce à l'axiome du choix dépendant. En utilisant l'axiome du choix dépendant sur

les individus du modèle, on peut se ramener à réaliser quel que soit le prédicat R la formule

$$ext(F), \forall R' \exists x F(R'x, R'\varphi(x)) \to \exists x F(Rx, R\psi(x))$$

pour une certaine application ψ définie dans le modèle. Il suffit en effet de prendre pour $\psi(x)$ un individu y tel que $\neg F(Rx, Ry)$, si un tel individu existe, et n'importe quel individu sinon.
On définit alors un prédicat R' par $R'\varphi^n(z) = R\psi^n(z)$ pour tout élément z ne possédant pas d'antécédent relativement à φ (ce qui définit R' sur tous les individus du modèle, celui-ci étant bien fondé pour φ). La deuxième hypothèse s'instancie alors en $\exists x F(R'x, R'\varphi(x))$. Considérant ensuite un élément $x = \varphi^n(z)$ (où z n'est pas un successeur) tel que l'on ait $F(R'x, R'\varphi(x))$, on peut conclure utilisant la formule $ext(F)$ et le fait que les formules $R'x$ et $R\psi^n(z)$ (resp. $R'\varphi(x)$ et $R'\psi^{n+1}(z)$) sont équivalentes.

L'effacement des quantificateurs universels positifs d'une formule correspond à la restriction des possibilités du joueur $\forall$ dans le jeu associé, puisque celui-ci est alors contraint de répondre fonctionnellement aux choix de son adversaire (Cf. l'exemple donné en 5.3). Considérant l'implémentation d'une stratégie gagnante pour un jeu donné dans lequel l'adversaire est supposé répondre fonctionnellement, la quasi-preuve associée au théorème considéré permet d'implémenter une stratégie gagnante contre tout adversaire (Cf. le théorème 5.2.4). Et ce indépendamment du jeu considéré, si tant est qu'on lui donne une description de la structure de la formule associée (c'est-à-dire des règles du jeu). Elle opère donc une opération très complexe et très générale, qui nécessite l'instruction γ. On verra comment il effectue cette transformation dans la section 7.4.

Afin d'éviter d'écrire une quasi-preuve trop complexe comme réalisateur du théorème de Herbrand, on va considérer de nouvelles instructions, comme cela a été fait pour réaliser l'axiome du choix dépendant. Celles-ci sont néanmoins facultatives, puisqu'on pourrait écrire des termes n'utilisant que les instructions cc et γ qui auraient les comportements décrits ci-contre. On considérera alors un ensemble $\perp\!\!\!\perp$ saturé pour toutes ces instructions.

On va donc considérer huit instructions, notées $F, F_0, F_1, h_0, H, h_1, H_0$ et H_1 dont on donne les règles d'exécution ci-dessous :

$$F \star \tau \cdot t \cdot \pi \succ t \star (F_0)\tau t \cdot (F_1)\tau t \cdot \pi$$

$$F_0 \star \tau \cdot t \cdot u \cdot \pi \succ u \star \lambda x \lambda y\ y \cdot ((F)\tau)\lambda u \lambda v(\tau)(h_0)tuv \cdot \pi$$

$$F_1 \star \tau \cdot t \cdot u \cdot \pi \succ u \star \lambda z((z)\lambda y(y)((F)\tau)\lambda u \lambda v(\tau)(h_0)tuv)\lambda g I \cdot \pi$$

$$h_0 \star t \cdot a \cdot b \cdot u_0 \cdot u_1 \cdot \pi \succ t \star \lambda s(\mathsf{cc})\lambda k(((s)\lambda x \lambda y\ x)I)\lambda c \lambda d(a)d(k)(u_0)c \cdot \lambda s(s)\lambda z((z)\lambda x \lambda y(x)(y)(u_1)I(b)I)I \cdot \pi$$

$$H \star p \cdot t \cdot \kappa \cdot \pi \succ t \star \lambda w((F)\lambda y(\kappa)(\gamma)y)\lambda a \lambda b(\kappa)(\gamma)(h_1)wpab \cdot \pi$$

$$h_1 \star w \cdot p \cdot a \cdot b \cdot u_0 \cdot u_1 \cdot \pi \succ p \star \lambda t(t)(b)I \cdot \lambda u \lambda v(a)vu \cdot (H_0)bu_1 \cdot (H_1)au_0 \cdot w \cdot \pi$$

$$H_0 \star b \cdot u_1 \cdot v \cdot \pi \succ v \star \lambda y((y)\lambda z(z)\lambda t(t)(u_1)I(b)I)I \cdot \pi$$

$$H_1 \star a \cdot u_0 \cdot v \cdot v' \cdot \pi \succ v' \star \lambda x \lambda y\ x \cdot I \cdot \lambda x \lambda y(a)y(\mathsf{k}_\pi)(u_0)xv \cdot \pi$$

7.3.2 Réalisation du théorème

Notations : On fixe jusqu'à la fin du chapitre une formule $F(U, V)$, où U et V sont des variables propositionnelles.
L'expression $\exists^* x[A, B]$ dénotera la formule $\forall x(A, B \to \bot) \to \bot$.

Lemme 7.3.1.
Pour tout prédicat 2-aire V, il existe un prédicat $S : \mathscr{N} \times \mathscr{N} \to \mathscr{P}(\Pi)$ tel que, quels que soient les éléments n et p de $\mathscr{N}$:

$$S(n,p) = \left|\left|\forall x' \left(\varphi(x') = n \to \exists^* y'[V(y',p), S(x',y')]\right) \bigwedge \left(n \simeq a \to p = n\right)\right|\right|.$$

La signification de ce prédicat est la suivante : considérant un individu $\varphi^k a'$ d'un modèle générique $\mathscr{M}$ tel que a' n'est pas dans l'image de φ, $\mathscr{M}$ satisfait $S(\varphi^k a', y)$ ssi il existe une suite $y_0 = a', y_2, \ldots, y_{k-1}, y_k = y$ telle que $V(y_i, y_{i+1})$ et $S(\varphi^i a', y_i)$ pour tout i.

Démonstration. Il suffit de constater que le symbole S ne possède que des occurrences positives dans la formule

$$\forall x' \left(\varphi(x') = n \to \exists^* y' [V(y', p), S(x', y')] \right) \bigwedge \left(n \simeq a \to p = n \right)$$

et d'utiliser le théorème 2.2.40. □

On fixe pour le lemme suivant un prédicat 2-aire V, ainsi qu'un prédicat S qui est l'un de ceux associés à V par le lemme ci-dessus. Il n'est a priori pas unique, mais l'on peut toujours choisir celui dont la valeur de vérité est la plus petite, car c'est celui qui aura « le plus de réalisateur », c'est-à-dire qu'intuitivement il ne spécifie rien d'autre que l'équivalence le définissant.

On commence par réaliser une formule exprimant le fait que le prédicat S est fonctionnel par rapport à son premier argument si V l'est (prendre $\Phi = \bot$).

Théorème 7.3.2.
Soit Φ une formule avec au plus une variable libre, et τ un terme réalisant la formule

$$\forall y \left[\forall \bar{z} \left(\forall \bar{y} \{ V(y, \bar{y}) \to \bar{y} = \bar{z} \}, \forall \bar{y} \{ \bar{y} = \bar{z} \to V(y, \bar{y}) \} \to \Phi(\bar{z}) \right) \to \bot \right].$$

Alors le terme $(F)\tau$ réalise

$$\forall x \left[\forall z \left(\forall y \{ S(x, y) \to y = z \}, \forall y \{ y = z \to S(x, y) \} \to \bot \right) \to \bot \right].$$

Il nous faut d'abord démontrer le lemme suivant qui exprime que si V est un prédicat fonctionnel par rapport à son premier argument, alors pour tout élément x, $S(\varphi(x), .)$ est un singleton si $S(x, .)$ l'est.

Lemme 7.3.3.
Avec l'hypothèse et les notations de l'énoncé précédent, la quasi-preuve $\lambda t \lambda u \lambda v (\tau)(h_0) t u v$ réalise la formule

$$\forall x \left(\forall z \left[\forall y \{ S(\varphi(x), y) \to y = z \}, \forall y \{ y = z \to S(\varphi(x), y) \} \to \bot \right] \to \forall z \left[\forall y \{ S(x, y) \to y = z \}, \forall y \{ y = z \to S(x, y) \} \to \bot \right] \right).$$

Démonstration. On commence par fixer $x \in \mathcal{N}$, un terme t réalisant $\forall z \left[\forall y \{ S(\varphi(x), y) \to y = z \}, \forall y \{ y = z \to S(\varphi(x), y) \} \to \bot \right]$, ainsi que $z \in \mathcal{N}$ et deux termes a et b réalisant respectivement $\forall y \{ S(x, y) \to y = z \}$ et $\forall y \{ y = z \to S(x, y) \}$.

Il nous faut démontrer que le processus $\lambda t \lambda u \lambda v (\tau)(h_0)tuv \star t \cdot a \cdot b \cdot \pi$ est dans $\perp\!\!\!\perp$ quelle que soit la pile π. Ce processus se réduisant en $\tau \star (h_0)tab \cdot \pi$, il suffit considérant l'hypothèse faite sur τ de vérifier que $(h_0)tab$ réalise la formule suivante :

$$\forall \bar{z} \left(\forall \bar{y}\{V(z,\bar{y}) \to \bar{y} = \bar{z}\}, \forall \bar{y}\{\bar{y} = \bar{z} \to V(z,\bar{y})\} \to \Phi(\bar{z},y) \right).$$

On fixe pour cela $\bar{z} \in \mathscr{N}$, une pile ρ dans $||\Phi(\bar{z})||$, et l'on considère deux termes u_0 et u_1 réalisant respectivement les formules $\forall \bar{y}\{V(z,\bar{y}) \to \bar{y} = \bar{z}\}$ et $\forall \bar{y}\{\bar{y} = \bar{z} \to V(z,\bar{y})\}$. On doit alors montrer que le processus $(h_0)tab \star u_0 \cdot u_1 \cdot \rho$ est dans $\perp\!\!\!\perp$. Mais celui-ci se réduisant en

$$t \star \lambda s(\mathsf{cc})\lambda k(((s)\lambda x \lambda y\, x)I)\lambda c \lambda d(a)d(k)(u_0)c \cdot \lambda s(s)\lambda z((z)\lambda x \lambda y(x)(y)(u_1)I(b)I)I \cdot \rho,$$

il nous suffit considérant l'hypothèse faite sur t de prouver que :

i) $\lambda s(\mathsf{cc})\lambda k(((s)\lambda x \lambda y\, x)I)\lambda c \lambda d(a)d(k)(u_0)c \Vdash \forall y\{S(\varphi(x),y) \to y = \bar{z}\}$;

ii) $\lambda s(s)\lambda z((z)\lambda x \lambda y(x)(y)(u_1)I(b)I)I \Vdash \forall y\{y = \bar{z} \to S(\varphi(x),y)\}$.

Pour prouver *i)*, on fixe $y \in \mathscr{N}$, ainsi qu'un terme ξ réalisant $S(\varphi(x),y)$ et une pile ρ' dans $||y = \bar{z}||$. On considère alors le processus $\lambda s(\mathsf{cc})\lambda k(((s)\lambda x \lambda y\, x)I)\lambda c \lambda d(a)d(\mathsf{k}_{\rho'})(u_0)c \star \xi \cdot \rho'$, qui se réduit en $\xi \star \lambda x \lambda y\, x \cdot I \cdot \lambda c \lambda d(a)d(\mathsf{k}_{\rho'})(u_0)c \cdot \rho'$. Mais puisque $S(\varphi(x),y)$ est la valeur de vérité d'une conjonction et $\lambda x \lambda y\, x$ réalise $\forall X \forall Y(X, Y \to X)$, il suffit de démontrer que la pile $I \cdot \lambda c \lambda d(a)d(\mathsf{k}_{\rho'})(u_0)c \cdot \rho'$ appartient à

$$\left|\left| \forall x' \left(\varphi(x') = \varphi(x), \forall y'(V(y',y), S(x',y') \to \perp) \to \perp \right) \right|\right|.$$

Or $I \Vdash \varphi(x) = \varphi(x)$, il nous suffit donc de prouver que $\lambda c \lambda d(a)d(\mathsf{k}_{\rho'})(u_0)c$ réalise $\forall y'(V(y',y), S(x,y') \to \perp)$. On fixe pour cela $y' \in \mathscr{N}$, ainsi que des termes v et v' réalisant respectivement les formules $V(y',y)$ et $S(x,y')$, ainsi qu'une pile ρ''. Il nous faut alors montrer que le processus $\lambda c \lambda d(a)d(\mathsf{k}_{\rho'})(u_0)c \star v \cdot v' \cdot \rho''$ est dans $\perp\!\!\!\perp$. Celui-ci se réduit en $a \star v' \cdot (\mathsf{k}_{\rho'})(u_0)v \cdot \rho''$, avec a qui réalise $S(x,y') \to y' = z$. *i)* découle alors directement du fait que $(\mathsf{k}_{\rho'})(u_0)v \cdot \rho'' \in ||y' = z||$. En effet, si y' et z sont des éléments différents de $\mathscr{N}$ il n'y a rien à démontrer ; sinon on a $u_0 \Vdash V(z,y) \to y = \bar{z}$, $v \Vdash V(z,y)$ et $\mathsf{k}_{\rho'} \Vdash y = \bar{z} \to \perp$ ce qui assure $(\mathsf{k}_{\rho'})(u_0)v \Vdash \perp$.

Pour prouver *ii)*, on fixe $y \in \mathscr{N}$ ainsi qu'un terme ξ réalisant $y = \bar{z}$ et une pile $\rho \in S(\varphi(x),y)$. Il faut montrer que le processus $\lambda s(s)\lambda z((z)\lambda x \lambda y(x)(y)(u_1)I(b)I)I \star \xi \cdot \rho$ est dans $\perp\!\!\!\perp$. Celui-ci se réduit en

$\xi \star \lambda z((z)\lambda x\lambda y(x)(y)(u_1)I(b)I)I \cdot \rho$. Si y est différent de $\bar{z}$ le résultat suit car alors $\xi \Vdash \top \to \bot$. Sinon, supposons que y et $\bar{z}$ sont égaux, et montrons $\lambda z((z)\lambda x\lambda y(x)(y)(u_1)I(b)I)I \Vdash S(\varphi(x), y)$. Soit alors $\Psi \in \mathscr{P}(\Pi)$, $\rho' \in \Psi$ et ν réalisant

$$\forall x' \left(\varphi(x') = \varphi(x) \to \exists^* y'[V(y', y), S(x', y')] \right) , \left(\varphi(x) \simeq a \to y = \varphi(x) \right) \to \Psi$$

Montrons que $\lambda z((z)\lambda x\lambda y(x)(y)(u_1)I(b)I)I \star \nu \cdot \rho'$ est dans $\perp\!\!\!\perp$. Ce processus se réduit en $\nu \star \lambda x\lambda y(x)(y)(u_1)I(b)I \cdot I \cdot \rho'$. Or $||\varphi(x) \simeq a|| = \bot$, donc il est clair que $I \Vdash \varphi(x) \simeq a \to y = \varphi(x)$. Il nous suffit alors de montrer que quels que soient $x' \in \mathscr{N}$, $\rho'' \in \Pi$, $\alpha \in |\varphi(x') = \varphi(x)|$ et $\beta \in |\forall y'(V(y', y), S(x', y') \to \bot)|$, le processus $\lambda x\lambda y(x)(y)(u_1)I(b)I \star \alpha \cdot \beta \cdot \rho''$ est dans $\perp\!\!\!\perp$. Ce processus se réduit en $\alpha \star (\beta)(u_1)I(b)I \cdot \rho''$, et le résultat suit donc immédiatement si x' et x sont différents. Sinon, il faut montrer que le processus $\beta \star (u_1)I(b)I \cdot \rho''$ est dans $\perp\!\!\!\perp$. Cela découle du fait que $(b)I \Vdash S(x, z)$ (car on vient de supposer que x et x' sont égaux), et que $(u_1)I \Vdash V(z, y)$ (car on avait supposé que y et $\bar{z}$ étaient égaux).

□

Un raisonnement par induction permet alors de démontrer le théorème 7.3.2.

Démonstration du théorème 7.3.2. On va prouver par induction sur x ($\mathscr{N}$ étant bien fondé pour φ) que quels que soient la pile π et le terme t réalisant $\forall z \left(\forall y\{S(x, y) \to y = z\}, \forall y\{y = z \to S(x, y)\} \to \bot \right)$, le processus $(F)\tau \star t \cdot \pi$ est dans $\perp\!\!\!\perp$.
On fixe donc $x \in \mathscr{N}$ ainsi qu'une pile π et un terme t de la sorte. Le processus $(F)\tau \star t \cdot \pi$ se réduisant en $t \star (F_0)\tau t \cdot (F_1)\tau t \cdot \pi$, il nous suffit considérant l'hypothèse faite sur t de montrer que :

i) $(F_0)\tau t \Vdash \forall y\{S(x, y) \to y = x\}$;
ii) $(F_1)\tau t \Vdash \forall y\{y = x \to S(x, y)\}$.

Pour prouver *i)*, on considère un élément y de $\mathscr{N}$, un terme u réalisant $S(x, y)$ ainsi qu'une pile ρ dans $||y = x||$. Il faut montrer $(F_0)\tau t \star u \cdot \rho \in \perp\!\!\!\perp$. Ce processus se réduit en $u \star \lambda x\lambda y\ y \cdot ((F)\tau)\lambda u\lambda v(\tau)(h_0)tuv \cdot \rho$. Mais comme $S(x, y)$ est la valeur de vérité d'une conjonction, et puisque $\lambda x\lambda y\ y \Vdash \forall X\forall Y(X, Y \to Y)$, il nous suffit pour avoir le résultat de montrer que $((F)\tau)\lambda u\lambda v(\tau)(h_0)tuv \cdot \rho$ est dans $||x \simeq a \to y = x||$. Si x est égal à a il n'y a rien à démontrer (car alors $||x \simeq a|| = \top$) ; sinon il faut montrer $((F)\tau)\lambda u\lambda v(\tau)(h_0)tuv \Vdash \bot$. Considérons alors un élément x' tel que $\varphi(x') = x$: il nous suffit grâce à l'hypothèse d'induction de montrer

$$\lambda u\lambda v(\tau)(h_0)tuv \Vdash \forall z \left(\forall y\{S(x', y) \to y = z\}, \forall y\{y = z \to S(x', y)\} \to \bot \right),$$

ce qui découle du lemme 7.3.3.

Pour démontrer *ii)*, on fixe $y \in \mathscr{N}$ ainsi qu'un terme u réalisant $y = x$ et une pile ρ dans $S(x,y)$. Il faut alors montrer que le processus $(F_1)\tau t \star u \cdot \rho$ est dans $\perp\!\!\!\perp$. Celui-ci se réduisant en $u \star \lambda z((z)\lambda y(y)((F)\tau)\lambda u\lambda v(\tau)(h_0)tuv)\lambda gI \cdot \rho$, on peut supposer que y et x sont égaux, car sinon le résultat est trivial. Il nous faut alors montrer que $\lambda z((z)\lambda y(y)((F)\tau)\lambda u\lambda v(\tau)(h_0)tuv)\lambda gI \Vdash S(x,x)$. Soit alors $\Psi \in \mathscr{P}(\Pi)$, $\rho' \in \Psi$ et ν réalisant

$$\forall x' \left(\varphi(x') = x \to \exists^* y'[V(y',x), S(x',y')]\right), \left(x \simeq a \to x = x\right) \to \Psi.$$

Montrons que $\lambda z((z)\lambda y(y)((F)\tau)\lambda u\lambda v(\tau)(h_0)tuv)\lambda gI \star \nu \cdot \rho'$ est dans $\perp\!\!\!\perp$. Ce processus se réduit en $\nu \star \lambda y(y)((F)\tau)\lambda u\lambda v(\tau)(h_0)tuv \cdot \lambda gI \cdot \rho'$, et puisque $\lambda g\ I$ réalise $x \simeq a \to x = x$, il nous suffit pour avoir le résultat de démontrer que $\lambda y(y)((F)\tau)\lambda u\lambda v(\tau)(h_0)tuv$ réalise $\forall x'[\varphi(x') = x \to \exists^* y'(V(y',x), S(x',y'))]$. Soient alors $x' \in \mathscr{N}$, ξ un terme réalisant $\varphi(x') = x$, ρ'' une pile dans $||\forall y'(V(y',x), S(x',y') \to \perp) \to \perp||$. On doit montrer que le processus $\lambda y(y)((F)\tau)\lambda u\lambda v(\tau)(h_0)tuv \star \xi \cdot \rho''$ est dans $\perp\!\!\!\perp$. Comme il se réduit en $\xi \star ((F)\tau)\lambda u\lambda v(\tau)(h_0)tuv \cdot \rho''$, on peut donc supposer $\varphi(x') = x$. Mais alors l'hypothèse d'induction et le lemme 7.3.3 assurent que $((F)\tau)\lambda u\lambda v(\tau)(h_0)tuv$ réalise $\perp$, ce qui prouve *ii)* et achève la démonstration du théorème. □

Enfin, on peut réaliser la deuxième partie du théorème de Herbrand.

Théorème 7.3.4.
Quelle que soit la formule $F(U,V)$ où U et V désignent des variables propositionnelles, la formule

$$ext(F), \forall R\exists^* xF(Rx, R\varphi(x)) \to \forall R\exists^* x\forall yF(Rx,Ry)$$

est réalisée par l'instruction H.

Démonstration. Soient e et t sont des termes réalisant respectivement les formules $ext(F)$ et $\forall R\exists^* xF(Rx, R\varphi(x))$, R un élément de $\mathscr{P}(\Pi)^{\mathscr{N}}$, u un terme réalisant $\forall x(\forall yF(Rx,Ry) \to \perp)$ et π une pile. Il nous faut montrer que le processus $p = H \star e \cdot t \cdot u \cdot \pi$ est dans $\perp\!\!\!\perp$. γ étant l'instruction définie en 6.2, elle réalise pour un certain prédicat V la formule suivante (voir le théorème 6.2.1) :

$$\forall x \left[\forall z \left(\forall y\{V(x,y) \to y = z\}, \forall y\{y = z \to V(x,y)\} \to F(Rx,Rz)\right) \to \forall yF(Rx,Ry)\right].$$

On désigne alors par S un prédicat associé à V par le lemme 7.3.1, et on appelle R' l'élément de $\mathscr{P}(\Pi)^{\mathscr{N}}$ défini par

$$\forall x \in \mathscr{N}, R'(x) = ||\forall \bar{x}[S(x,\bar{x}) \to R(\bar{x})]||.$$

Considérant alors que t réalise $\forall x(F(R'x, R'\varphi(x)) \to \perp) \to \perp$ et que le processus p se réduit en $t \star \lambda w((F)\lambda y(u)(\gamma)y)\lambda a\lambda b(u)(\gamma)(h_1)weab \cdot \pi$, il nous suffit pour avoir le résultat de démontrer que

$$\lambda w((F)\lambda y(u)(\gamma)y)\lambda a\lambda b(u)(\gamma)(h_1)weab \Vdash \forall x(F(R'x, R'\varphi(x)) \to \perp).$$

Fixons donc un élément x de $\mathscr{N}$ et un terme w réalisant $F(R'x, R'\varphi(x))$; considérant le théorème 7.3.2 il nous suffit de démontrer que :

i) $\lambda y(u)(\gamma)y$ réalise

$$\forall y \left[\forall \bar{z} \left(\forall \bar{y}\{V(y,\bar{y}) \to \bar{y} = \bar{z}\}, \forall \bar{y}\{\bar{y} = \bar{z} \to V(y,\bar{y})\} \to F(Rx, R\bar{z}) \right) \to \bot \right] ;$$

ii) $\lambda a \lambda b(u)(\gamma)(h_1)weab$ réalise

$$\forall z \left(\forall y\{S(x,y) \to y = z\}, \forall y\{y = z \to S(x,y)\} \to \bot \right).$$

Pour *i*), on fixe $y \in \mathscr{N}$, un terme ξ réalisant

$$\forall \bar{z} \left(\forall \bar{y}\{V(y,\bar{y}) \to \bar{y} = \bar{z}\}, \forall \bar{y}\{\bar{y} = \bar{z} \to V(y,\bar{y})\} \to F(Rx, R\bar{z}) \right)$$

et on doit montrer que le processus $\lambda y(u)(\gamma)y \star \xi \cdot \rho$ est dans $\perp\!\!\!\perp$ pour toute pile ρ. Ce processus se réduit en $u \star (\gamma)\xi \cdot \rho$, et considérant la formule que réalise γ, on a que $(\gamma)\xi$ réalise $\forall y F(Rx, Ry)$, ce qui donne le résultat.

Il nous reste donc à montrer l'assertion *ii*). On fixe pour cela $z \in \mathscr{N}$, deux termes α et β réalisant respectivement les formules $\forall y\{S(x,y) \to y = z\}$ et $\forall y\{y = z \to S(x,y)\}$. On doit montrer que $\lambda a \lambda b(u)(\gamma)(h_1)weab \star \alpha \cdot \beta \cdot \rho \in \perp\!\!\!\perp$ pour toute pile ρ. Ce processus se réduisant en $u \star (\gamma)(h_1)we\alpha\beta \cdot \rho$ il suffit, considérant la formule réalisée par γ, de démontrer que $(h_1)we\alpha\beta$ réalise

$$\forall \bar{z} \left(\forall \bar{y}\{V(z,\bar{y}) \to \bar{y} = \bar{z}\}, \forall \bar{y}\{\bar{y} = \bar{z} \to V(z,\bar{y})\} \to F(Rz, R\bar{z}) \right).$$

On fixe donc $\bar{z} \in \mathscr{N}$, deux termes u_0 et u_1 réalisant respectivement $\forall \bar{y}\{V(z,\bar{y}) \to \bar{y} = \bar{z}\}$ et $\forall \bar{y}\{\bar{y} = \bar{z} \to V(z,\bar{y})\}$, ainsi qu'une pile ρ' dans $||F(Rz, R\bar{z})||$, et l'on doit montrer que le processus $(h_1)we\alpha\beta \star u_0 \cdot u_1 \cdot \rho'$ est dans $\perp\!\!\!\perp$. Celui-ci se réduisant en $e \star \lambda t(t)(\beta)I \cdot \lambda u \lambda v(\alpha)vu \cdot (H_0)\beta u_1 \cdot (H_1)\alpha u_0 \cdot w \cdot \rho'$ avec e qui réalise :

$$(R'x \to Rz), (Rz \to R'x), (R'\varphi x \to R\bar{z}), (R\bar{z} \to R'\varphi x), F(R'x, R'\varphi x) \to F(Rz, R\bar{z}),$$

il ne nous reste plus qu'à démontrer :

j) $\lambda t(t)(\beta)I \Vdash R'x \to Rz$;
jj) $\lambda u \lambda v(\alpha)vu \Vdash Rz \to R'x$;
jjj) $(H_0)\beta u_1 \Vdash R'\varphi(x) \to R\bar{z}$;
jv) $(H_1)\alpha u_0 \Vdash R\bar{z} \to R'\varphi(x)$.

Pour j), on considère un terme t réalisant $R'x$ et une pile $\bar{\rho}$ dans Rz. Il faut alors montrer que le processus $\lambda t(t)(\beta)I \star t \cdot \bar{\rho}$ est dans $\perp\!\!\!\perp$, ce qui découle du fait qu'il se réduit en $t \star (\beta)I \cdot \bar{\rho}$, avec t qui réalise $S(x,z) \to Rz$ et $(\beta)I$ qui réalise $S(x,z)$.

Pour jj), on considère un terme ξ réalisant $R(z)$, $\bar{x} \in \mathscr{N}$, un terme ξ' réalisant $S(x,\bar{x})$ et une pile $\bar{\rho}$ dans $R\bar{x}$. Il faut alors montrer $\lambda u \lambda v(\alpha)vu \star \xi \cdot \xi' \cdot \bar{\rho} \in \perp\!\!\!\perp$. Or celui-ci se réduit en $(\alpha)\xi' \star \xi \cdot \bar{\rho}$, avec $(\alpha)\xi'$ qui réalise $\bar{x} = z$. Si $\bar{x} \neq z$, le résultat est démontré car alors $\xi \cdot \bar{\rho} \in ||\top \to \bot|| = ||\bar{x} = z||$; sinon, le résultat découle du fait que $\xi \star \bar{\rho} \in \perp\!\!\!\perp$ et donc que $\xi \cdot \bar{\rho} \in ||\forall X(X \to X)|| = ||\bar{x} = z||$.

Pour jjj), On considère un terme ξ réalisant $R'\varphi(x)$ ainsi qu'une pile $\bar{\rho}$ dans $R\bar{z}$, et on doit montrer que le processus $H_0 \star \beta \cdot u_1 \cdot \xi \cdot \bar{\rho}$ est dans $\perp\!\!\!\perp$. Celui-ci se réduisant en $\xi \star \lambda y((y)\lambda z(z)\lambda t(t)(u_1)I(\beta)I)I \cdot \bar{\rho}$, il suffit de démontrer que $\lambda y((y)\lambda z(z)\lambda t(t)(u_1)I(\beta)I)I$ réalise $S(\varphi(x),\bar{z})$. Mais il est clair que I réalise $\varphi(x) \simeq a \to \varphi(x) = \bar{z}$ car $||\varphi(x) \simeq a|| = \bot$. Il ne reste donc qu'à montrer que $\lambda z(z)\lambda t(t)(u_1)I(\beta)I$ réalise :

$$\forall x' \left(\varphi(x') = \varphi(x), \forall y'(V(y',\bar{z}), S(x,y') \to \bot) \to \bot \right).$$

On fixe alors $x' \in \mathscr{N}$, deux termes ν et ν' réalisant respectivement $\varphi(x') = \varphi(x)$ et $\forall y'(V(y',\bar{z}), S(x,y') \to \bot)$ ainsi qu'une pile $\bar{\pi}$, et il nous faut démontrer que le processus $\lambda z(z)\lambda t(t)(u_1)I(\beta)I \star \nu \cdot \nu' \cdot \bar{\pi}$ est dans $\perp\!\!\!\perp$. Ce processus se réduit en $\nu \star \lambda t(t)(u_1)I(\beta)I \cdot \nu' \cdot \bar{\pi}$, ce qui prouve le résultat si $x' \neq x$, car alors $\nu \Vdash \top \to \bot$. Sinon, il reste à démontrer que le processus $\lambda t(t)(u_1)I(\beta)I \star \nu' \cdot \bar{\pi}$ est dans $\perp\!\!\!\perp$; mais celui-ci se réduit en $\nu' \star (u_1)I \cdot (\beta)I \cdot \bar{\pi}$ avec $(u_1)I$ qui réalise $V(z,\bar{z})$ et $(\beta)I$ qui réalise $S(x,z)$, ce qui donne le résultat.

Enfin, pour démontrer jv), on considère un terme ξ réalisant $R\bar{z}$, un élément $\bar{x} \in \mathscr{N}$, un terme ξ' réalisant $S(\varphi(x),\bar{x})$ ainsi qu'une pile $\bar{\rho} \in R\bar{x}$, et l'on doit montrer que le processus $H_1 \star \alpha \cdot u_0 \cdot \xi \cdot \xi' \cdot \bar{\rho}$ est dans $\perp\!\!\!\perp$. Ce processus se réduisant en $\xi' \star \lambda x \lambda y\, x \cdot I \cdot \lambda x \lambda y(\alpha)y(\mathsf{k}_{\bar{\rho}})(u_0)x\xi \cdot \bar{\rho}$, il suffit de démontrer, considérant que $S(\varphi(x),\bar{x})$ est la valeur de vérité d'une conjonction et que $\lambda x \lambda y\, x$ réalise $\forall X \forall Y(X, Y \to X)$, que la pile $I \cdot \lambda x \lambda y(\alpha)y(\mathsf{k}_{\bar{\rho}})(u_0)x\xi \cdot \bar{\rho}$ est dans

$$||\forall x'(\varphi(x') = \varphi(x) \to \exists^* y'[V(y',\bar{x}), S(x',y')])||.$$

Comme $I \Vdash \varphi(x) = \varphi(x)$, il reste à prouver que $\lambda x \lambda y(\alpha)y(\mathsf{k}_{\bar{\rho}})(u_0)x\xi$ réalise $\forall y'(V(y',\bar{x}), S(x,y') \to \bot)$. On fixe donc $y' \in \mathscr{N}$, ainsi que des termes ν et ν' réalisant respectivement $V(y',\bar{x})$ et $S(x,y')$, et il nous faut montrer que le processus $\lambda x \lambda y(\alpha)y(\mathsf{k}_{\bar{\rho}})(u_0)x\xi \star \nu \cdot \nu' \cdot \bar{\pi}$ est dans $\perp\!\!\!\perp$ pour toute pile $\bar{\pi}$. Or ce processus se réduit en $(\alpha)\nu' \star (\mathsf{k}_{\bar{\rho}})(u_0)\nu\xi \cdot \bar{\pi}$, avec $(\alpha)\nu'$ qui réalise $y' = z$. On peut donc supposer que y' et z sont égaux, et il reste à démontrer que dans ce cas $(\mathsf{k}_{\bar{\rho}})(u_0)\nu \star \xi \cdot \bar{\pi}$ est dans $\perp\!\!\!\perp$. Mais ce processus se réduit en $(u_0)\nu\xi \star \bar{\rho}$, avec $(u_0)\nu$ qui réalise $\bar{x} = \bar{z}$, puisqu'on a supposé que y' et z étaient égaux. On peut donc supposer que $\bar{x} = \bar{z}$, et il faut alors montrer que dans ce cas le processus $\xi \star \bar{\rho}$ est dans $\perp\!\!\!\perp$, ce qui découle des hypothèses.

□

7.4 Analyse du terme obtenu

Notations : On appelle $F(U,V)$ la formule $U \to V$.
Soit $\epsilon = \lambda a\lambda b\lambda c\lambda d\lambda e\lambda f(c)(e)(b)f$, qui réalise $ext(F)$.
Soit $\xi = \lambda z(z)\lambda x(z)\lambda g\ x$, le terme réalisant la formule $\forall R\exists x F(Rx, R\varphi(x))$ donné au théorème 5.3.3.
κ_h désigne la constante d'interaction associée à la formule $\exists x F(Rx, R\varphi x)$, et κ celle associée à la formule $\exists x\forall y F(Rx, Ry)$ (Cf. la section 5.3). On notera alors κ_i la constante d'interaction associée à la formule Ri, et π_j la constante de pile associée à la formule Rj. On note enfin i_p (resp. j_p) le p-ième entier choisi par $\exists$ (resp. $\forall$) dans le jeu associé au processus $H \star \epsilon \cdot \xi \cdot \kappa \cdot \pi$.

Commençons par décrire l'exécution du processus $H \star \epsilon \cdot \xi \cdot \kappa \cdot \pi$. Les seules piles qui sont numérotées au cours de cette exécution par l'instruction γ sont celles de la forme $\kappa_{i_p} \cdot \pi_{j_p}$; on notera n_p le numéro de cette pile. L'exécution fait d'abord apparaître κ en tête six fois, c'est-à-dire que six tours de jeu ont lieu. Les entiers n_3 et n_5 sont alors comparés par l'instruction U_1. La suite de l'exécution dépend de l'ordre relatif entre ces deux entiers :
Si $n_3 < n_5$, l'exécution se termine sur le processus $\kappa_{i_4} \star \pi_{j_3}$.
Si $n_3 = n_5$, c'est-à-dire si $i_3 = i_5$ et $j_3 = j_5$, l'exécution se termine sur le processus $\kappa_{i_4} \star \pi_{j_5}$.
Si $n_3 > n_5$, κ apparaît en tête deux fois de plus, puis U_1 compare les entiers n_5 et n_7. La suite de l'exécution dépend à nouveau de l'ordre relatif entre ces deux entiers, la discussion étant analogue à celle ci-dessus :
Si $n_5 < n_7$, l'exécution se termine sur le processus $\kappa_{i_6} \star \pi_{j_5}$.
Si $n_5 = n_7$, c'est-à-dire si $i_5 = i_7$ et $j_5 = j_7$, l'exécution se termine sur le processus $\kappa_{i_6} \star \pi_{j_7}$.
Si $n_5 > n_7$, κ apparaît en tête deux fois de plus, puis U_1 compare les entiers n_5 et n_7...
Il ne reste plus qu'à remarquer que l'exécution termine en temps fini puisque l'ordre mis sur les piles est bien fondé : la suite (n_{2p+1}) ne peut donc décroître indéfiniment. La figure 7.1 schématise cette description, l'axe des abscisses faisant apparaître les tours de jeu, repérés par leur numéro.

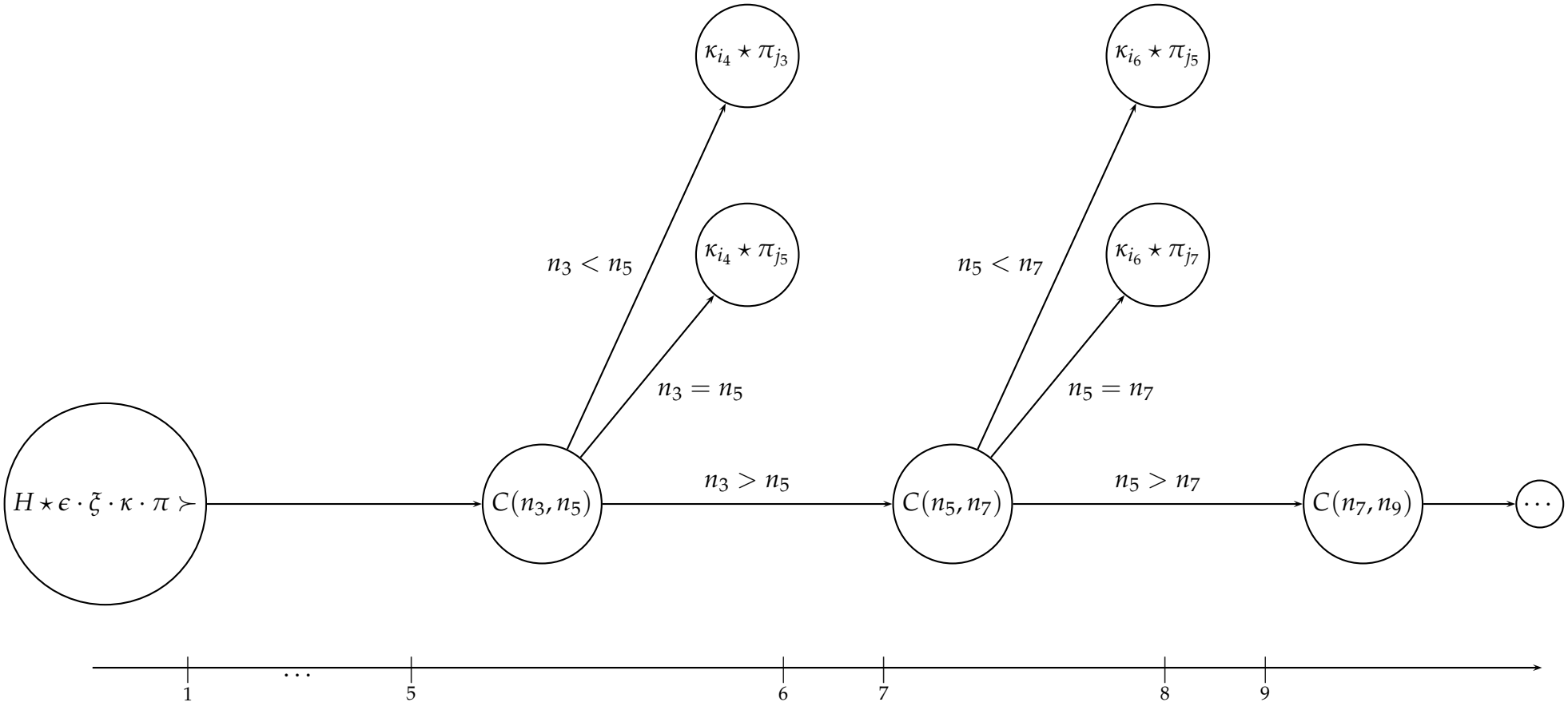


FIG. 7.1 – Chronogramme des exécutions possibles de $H \star \epsilon \cdot \xi \cdot \kappa \cdot \pi$.

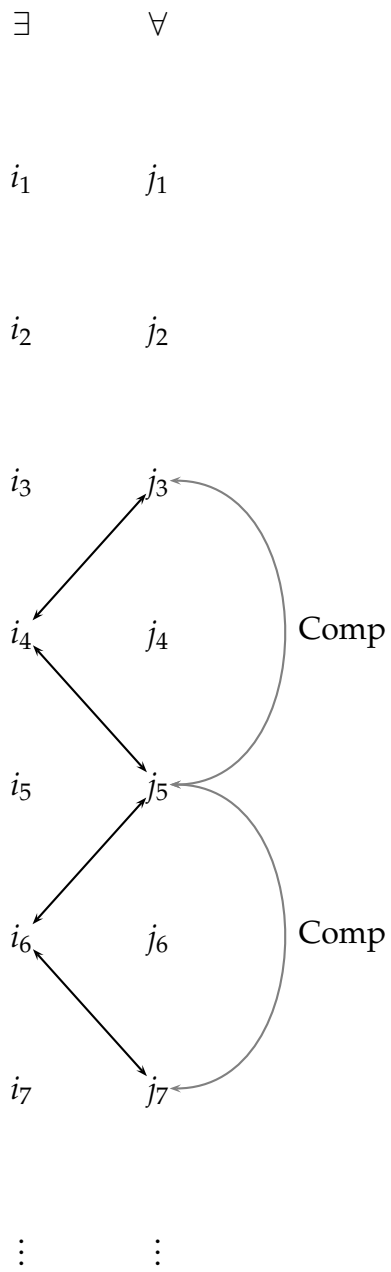


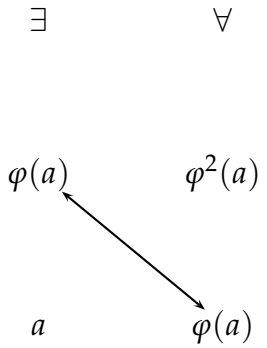
FIG. 7.2 – Trace de la partie associée à l'exécution du processus $H \star \epsilon \cdot \xi \cdot \kappa \cdot \pi$.

Analyse de la stratégie de $\exists$ dans ce cas : La figure 7.2 représente la trace de la partie associée à cette exécution. On y indique les coups joués par $\exists$ et $\forall$; les doubles flèches rectilignes indiquant les égalités conditionnant l'issue de la partie (suivant l'ordre mis sur les piles), et les flèches curvilignes quels sont les

couples comparés (via les instructions et piles associés) par l'instruction U_1.

Etant donné une partie (i_n, j_n), $\exists$ est certaine de l'emporter quel que soit la numérotation des piles si $i_{2(n+1)} = j_{2n+1}$ pour tout $n \geqslant 1$. C'est-à-dire qu'une stratégie gagnante pour $\exists$ indépendante de l'ordre sur les piles est de toujours jouer au $2(n+1)$-ième coup ce qu'a joué $\forall$ au précédent.

On peut considérer que H effectue sur une stratégie gagnante contre κ_h une opération permettant d'obtenir une stratégie gagnante contre l'instruction κ. Nous allons voir que dans le cas de la quasi-preuve ξ considérée, cette opération s'interprète naturellement ; l'instruction H correspondant à n'importe quel protocole de communication (modulo un terme décrivant la formule F choisie, c'est-à-dire le protocole considéré), son action sur les stratégies doit donc être la plus générale possible. Dans le processus $\xi \star \kappa_h \cdot \pi$, la stratégie gagnante pour $\exists$ est de jouer $\varphi(a)$ puis a, ce qui donne la partie suivante :



Dans le processus $H \star \epsilon \cdot \xi \cdot \kappa \cdot \pi$, $\exists$ ne peut évidemment pas se contenter de reproduire cette stratégie, puisque la constante d'interaction κ n'est pas fonctionnelle. Il n'existe pas de fonction jouant le rôle de φ pour κ, puisque le coup joué par $\forall$ à un instant donné ne dépend pas qu'exclusivement du coup joué par $\exists$ au même moment. Appelons donc $\psi_k(x)$ l'entier joué par $\forall$ au k-ième tour de jeu lorsque $\exists$ joue x à ce même tour. La nouvelle stratégie gagnante de $\exists$ va correspondre à l'adaptation la plus simple de l'ancienne.

$\exists$ ne pouvant commencer par jouer $\varphi(a)$ (puisque cet élément est censé être la réponse de l'opposant à la question a, qu'il ne connaît pas) elle commence par jouer a, ce à quoi $\forall$ répond $\psi_1(a)$. $\exists$ peut alors recopier la stratégie gagnante associée au réalisateur de la forme de Herbrand considéré, c'est-à-dire qu'elle joue ensuite $\psi_1(a)$ puis a. $\forall$ répond alors par $\psi_2\psi_1(a)$ puis $\psi_3(a)$. Si $\psi_3(a) = \psi_1(a)$, c'est-à-dire si $\forall$ donne alors la même réponse à a que celle qu'il avait donné lors du premier tour, $\exists$ l'emporte. Sinon, $\exists$ est obligée d'appliquer à nouveau l'ancienne stratégie : elle joue alors $\psi_3(a)$ puis a. $\exists$ réitère sans cesse ce procédé, et l'emporte si au cours de la partie $\forall$ donne deux fois la même réponse à la question a.

La figure 7.3 donne la trace de la partie lorsque $\exists$ applique cette stratégie.

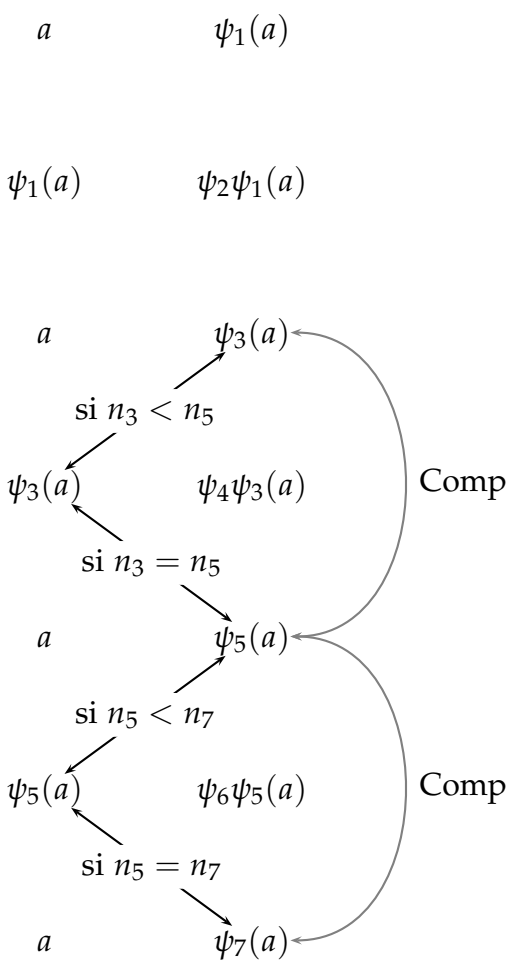


FIG. 7.3 – Trace de la partie associée à $\xi \star \kappa_h \cdot \pi$ lorsque $\exists$ applique la stratégie énoncée.

On voit que lorsque $\exists$ applique cette stratégie dans la partie associée au processus $\xi \star \kappa_h \cdot \pi$, elle l'emporte au moment où se produit l'un des deux phénomènes suivants :

- $\forall$ donne deux fois de suite la même réponse à la question a ; alors $\exists$ l'emporte avec un coup gagnant correspondant à un coup ultérieur de $\forall$, c'est-à-dire dans des conditions similaires à celles des parties associées au processus $\xi \star \kappa_h \cdot \pi$.
- il existe un entier p tel que $n_{2p+3} > n_{2p+1}$, auquel cas $\exists$ l'emporte avec un coup gagnant correspondant à un coup antérieur de $\forall$.

Cette stratégie est donc gagnante pour $\exists$ dans les parties associées au processus $H \star \epsilon \cdot \xi \cdot \kappa \cdot \pi$, puisque la bonne fondation de l'ordre sur les piles empêche la suite d'entiers (n_{2p+1}) de décroître indéfiniment.

On remarque également que lorsque $\exists$ applique cette stratégie, les entiers comparés par U_1 sont ceux associés aux tours où $\exists$ joue a : ces comparaisons s'interprètent donc comme permettant de savoir si $\forall$ a donné deux fois consécutives la même réponse à la question a, c'est-à-dire comme testant la fonctionnalité du comportement de l'opposant. La numérotation des piles n'est utilisée que pour ce faire.

La démonstration usuelle de l'application considérée du théorème de Herbrand correspond à la construction d'une fonction de choix dans un modèle afin d'interpréter le symbole de fonction φ. De manière analogue, tout se déroule ici comme si H permettait à $\exists$ de construire un élément $\psi(a)$ à choisir parmi toutes les réponses $\psi_k(a)$ données par l'opposant à la question a ; le choix s'effectuant selon le principe suivant :

« Si $\forall$ donne à un moment une réponse supérieure à la précédente (au sens de l'ordre mis sur les piles), l'avant-dernière réponse est choisie. »

Interprétation dans le cadre des protocoles de communication : la formule $\forall R \exists x \forall y (Rx \rightarrow Ry)$ peut être interprétée comme spécifiant un protocole permettant l'envoi d'un paquet (Cf. [28]). La forme de Herbrand de celle-ci permet manifestement de modéliser des sessions où la connexion est défaillante : dans la partie associée au processus $\lambda z(z)\lambda x(z)\lambda g x \star \kappa_h \cdot \pi$, le serveur accepte un acquittement relatif à un message émis ultérieurement. L'action de H sur ce terme peut être interprétée comme la détection de la défaillance de la connexion puis l'application d'une stratégie de correction ; l'ordre sur les piles déterminant le temps nécessaire à ce que la session soit finalement couronnée de succès.

Conclusion

Soulignons enfin certaines des questions soulevées par cette thèse, qui sont autant de perspectives de recherches futures.

Les entiers comme classes d'équivalences. La méthode de raisonnement donnée dans le chapitre 3 est très générale puisqu'elle peut être adaptée à un modèle de ZF, et permet d'y définir les entiers. Cependant, les spécifications présentées ici ne sont pas en l'état très intéressantes. Les programmes obtenus effectuent des opérations sur les uplets d'entiers, qui deviennent rapidement incompréhensibles : la spécification associée à une formule exprimant l'associativité de l'addition induite semble déjà inutilisable tant le calcul mené est particulier. Il est probable que celles-ci possèdent une interprétation précise, mais dans un cadre qui reste à déterminer. L'étude de cette méthode dans un modèle de ZF nous éclairerait peut-être sur ce point.

Jeux et spécification de protocoles réseaux. Les principes de la modélisation présentée dans ce chapitre semblent robustes, mais il reste encore de nombreux points à préciser sur le sujet. La spécification des protocoles existants mériterait également d'être poussée plus en avant. On peut par exemple se demander comment modéliser le principe de la fenêtre glissante.
C'est la dissymétrie existant entre les rôles impartis aux deux joueurs qui permet cette modélisation ; elle impose néanmoins des contraintes strictes sur les protocoles spécifiés : comment s'en accommoder ?

Les jeux en réalisabilité. L'intérêt des processus associés à une formule jouable n'est pas encore évident. L'exécution de l'un d'entre eux décrit une session bien précise, en ce sens que les messages qui arrivent à destination au cours de celle-ci sont déterminés de manière univoque. Quelle peut être leur utilité du point de vue de la programmation ? On aimerait a priori pouvoir programmer des serveurs respectant un protocole donné, mais le comportement de ceux-ci correspond aux instructions κ_Φ, a priori associées à des formules qui ne sont pas réalisables. Au contraire, un processus implémente le comportement du client. Que faut-il en penser ?

Le théorème de Herbrand. L'opération effectuée par l'instruction H n'est pas encore bien comprise, et son utilité l'est encore moins, notamment pour les raisons évoquées ci-dessus.
On ne voit par ailleurs pas encore comment spécifier la formule qu'elle réalise, et plus généralement le problème de la spécification pour les formules du second ordre reste à étudier. Peut-on encore associer un jeu à chacune d'elle ? Et si oui, à quoi correspondent-ils ?

Annexe A

Quelques preuves présentées en séquents

On donne ici la présentation en séquents de quelques démonstrations du chapitre 3. On commence par le théorème 3.1.19.

Théorème.
La formule $\forall x_1 \forall x_2 [(x_1 \sqsubseteq x_2 \to \bot), (x_2 \sqsubseteq x_1 \to \bot) \to \bot]$ *est réalisé par chacune des quasi-preuves suivantes :*

i) $(Y)\lambda r \lambda u \lambda v (u) \lambda x (v)(r)x$;

ii) $(Y)\lambda r \lambda u \lambda v (v) \lambda y (u) \lambda x (r) x y$.

On démontre d'abord le lemme suivant.

Lemme.
Soit $a, b \in \mathbb{N}$, *F une formule à paramètre possédant comme seule variable libre x. Alors :*
i) $||(\forall x (F(x) \to sx \neq a) \to \bot) \to \bot|| \supseteq ||\forall x \{((F(x) \to \bot) \to \bot) \to sx \neq a\}||$.
ii) $||(F(a) \to sa \neq b) \to sa \neq b|| \subseteq ||(F(a) \to \bot) \to \bot||$.

Démonstration. i) Si $a = 0$, on a $||\forall x ((\{F(x) \to \bot\} \to \bot) \to sx \neq a)|| = \top$, ce qui donne le résultat. Sinon, $||\forall x ((\{F(x) \to \bot\} \to \bot) \to sx \neq a)||$ est l'ensemble des piles de la forme $t.\pi$ avec $t \in |(F(a-1) \to \bot) \to \bot|$. Mais le lemme 2.2.44 assure que $|(F(a-1) \to \bot)| = |\forall x (F(x) \to sx \neq a)|$, ce qui prouve que $t \in |\forall x (F(x) \to sx \neq a) \to \bot|$ et termine la démonstration.
ii) Trivial. □

Démonstration du théorème 3.1.19. i) On utilise le corollaire 2.2.19, ce qui nous permet de démontrer la formule considérée par induction sur b. On dénote donc $F(b)$ la formule suivante :

$$\forall a [(a \sqsubseteq b \to \bot), (b \sqsubseteq a \to \bot) \to \bot].$$

Montrons déjà que $\lambda r \lambda u \lambda v (u) \lambda x (v)(r)x$ réalise $\forall b (F(b) \to F(sb))$. On prend les notations suivantes :

- R dénote le contexte $r : F(b)$,
- U dénote le contexte $u : a \sqsubseteq sb \to \bot$,
- V dénote le contexte $v : sb \sqsubseteq a \to \bot$,
- X dénote le contexte $x : \forall b' (a' \sqsubseteq b' \to sb' \neq sb)$.

La preuve commence comme suit :

$$\dfrac{\dfrac{\dfrac{X \vdash x : \forall b' (a' \sqsubseteq b' \to sb' \neq sb)}{X \vdash x : a' \sqsubseteq b \to sb \neq sb}}{X \vdash x : a' \sqsubseteq b \to \bot}\,\mathrm{R} \qquad \dfrac{R \vdash r : F(b)}{R \vdash r : (a' \sqsubseteq b \to \bot), (b \sqsubseteq a' \to \bot) \to \bot}}{X, R \vdash (r)x : (b \sqsubseteq a' \to \bot) \to \bot}$$

Mais on a :

$$\begin{aligned} ||sb \sqsubseteq a \to \bot|| &= ||\forall b' (\forall a' (b' \sqsubseteq a' \to sa' \neq a) \to sb' \neq sb) \to \bot|| \\ &= ||(\forall a' (b \sqsubseteq a' \to sa' \neq a) \to \bot) \to \bot|| \text{ par le lemme 2.2.44} \\ &= ||\forall a' (((b \sqsubseteq a' \to \bot) \to \bot) \to sa' \neq a)|| \text{ par le lemme précédent.} \end{aligned}$$

La preuve peut donc se continuer comme suit :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{X, R \vdash (r)x : (b \sqsubseteq a' \to \bot) \to \bot} \qquad \dfrac{V \vdash v : \forall a' (((b \sqsubseteq a' \to \bot) \to \bot) \to sa' \neq a)}{V \vdash v : ((b \sqsubseteq a' \to \bot) \to \bot) \to sa' \neq a}}{X, R, V \vdash (v)(r)x : sa' \neq a}}{R, V \vdash \lambda x (v)(r)x : \forall b' (a' \sqsubseteq b' \to sb' \neq sb) \to sa' \neq a}}{R, V \vdash \lambda x (v)(r)x : \forall a' (\forall b' (a' \sqsubseteq b' \to sb' \neq sb) \to sa' \neq a)}\,\mathrm{R}}{R, V \vdash \lambda x (v)(r)x : a \sqsubseteq sb}$$

On peut enfin conclure :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{R, V \vdash \lambda x (v)(r)x : a \sqsubseteq sb} \qquad U \vdash u : a \sqsubseteq sb \to \bot}{R, V, U \vdash (u)\lambda x (v)(r)x : \bot}}{R, U \vdash \lambda v (u) \lambda x (v)(r)x : (sb \sqsubseteq a \to \bot) \to \bot}}{R \vdash \lambda u \lambda v (u) \lambda x (v)(r)x : (a \sqsubseteq sb \to \bot), (sb \sqsubseteq a \to \bot) \to \bot}}{R \vdash \lambda u \lambda v (u) \lambda x (v)(r)x : F(sb)}}{\vdash \lambda r \lambda u \lambda v (u) \lambda x (v)(r)x : F(b) \to F(sb)}}{\vdash \lambda r \lambda u \lambda v (u) \lambda x (v)(r)x : \forall b (F(b) \to F(sb))}$$

Il reste à voir que $\lambda r\lambda u\lambda v(u)\lambda x(v)(r)x$ réalise $\top \to F(0)$. On prend alors les notations suivantes :

- R dénote le contexte $r : \top$,
- U dénote le contexte $u : a \sqsubseteq 0 \to \bot$,
- V dénote le contexte $v : 0 \sqsubseteq a \to \bot$,
- X dénote le contexte $x : \forall b'(a' \sqsubseteq b' \to sb' \neq 0)$.

La preuve est alors la suivante :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{R, X \vdash (r)x : \top} \quad \dfrac{}{V \vdash v : 0 \sqsubseteq a \to \bot}\,\mathrm{R}}{V \vdash v : \top \to \bot}}{R, X, V \vdash (v)(r)x : \bot}}{R, X, V \vdash (v)(r)x : sa' \neq a}}{R, V \vdash \lambda x(v)(r)x : \forall b'(a' \sqsubseteq b' \to sb' \neq 0) \to sa' \neq a}}{R, V \vdash \lambda x(v)(r)x : \forall a'(\forall b'(a' \sqsubseteq b' \to sb' \neq 0) \to sa' \neq a)}}{R, V \vdash \lambda x(v)(r)x : a \sqsubseteq 0}\,\mathrm{R} \quad U \vdash u : a \sqsubseteq 0 \to \bot}{R, V, U \vdash (u)\lambda x(v)(r)x : \bot}}{R, U \vdash \lambda v(u)\lambda x(v)(r)x : (0 \sqsubseteq a \to \bot) \to \bot}}{R \vdash \lambda u\lambda v(u)\lambda x(v)(r)x : (a \sqsubseteq 0 \to \bot), (0 \sqsubseteq a \to \bot) \to \bot}}{R \vdash \lambda u\lambda v(u)\lambda x(v)(r)x : F(0)}}{\vdash \lambda r\lambda u\lambda v(u)\lambda x(v)(r)x : \top \to F(0)}$$

ii) Le second terme correspond encore à une preuve par induction sur b. On note toujours $F(b)$ la formule suivante :

$$\forall a[(a \sqsubseteq b \to \bot), (b \sqsubseteq a \to \bot) \to \bot].$$

Montrons déjà que $\lambda r\lambda u\lambda v(v)\lambda y(u)\lambda x(r)xy$ réalise $\forall b(F(b) \to F(sb))$. On prend les notations suivantes :

- R dénote le contexte $r : F(b)$,
- U dénote le contexte $u : a \sqsubseteq sb \to \bot$,
- V dénote le contexte $v : sb \sqsubseteq a \to \bot$,
- Y dénote le contexte $y : \forall a'(b \sqsubseteq a' \to sa' \neq a)$,
- X dénote le contexte $x : \forall b'(a' \sqsubseteq b' \to sb' \neq sb)$.

La dérivation suivante donne le résultat, en vertu du lemme précédent.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{X \vdash x : \forall b'(a' \sqsubseteq b' \to sb' \neq sb)}}{X \vdash x : a' \sqsubseteq b \to sb \neq sb}\,\mathrm{R}}{X \vdash x : a' \sqsubseteq b \to \bot} \quad \dfrac{}{R \vdash r : F(b)}}{R \vdash r : (a' \sqsubseteq b \to \bot), (b \sqsubseteq a' \to \bot) \to \bot}}{X, R \vdash (r)x : (b \sqsubseteq a' \to \bot) \to \bot}}{X, R \vdash (r)x : (b \sqsubseteq a' \to sa' \neq a) \to sa' \neq a}\,\mathrm{R}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{X, R \vdash (r)x : (b \sqsubseteq a' \to sa' \neq a) \to sa' \neq a} \quad \dfrac{Y \vdash y : \forall a'(b \sqsubseteq a' \to sa' \neq a)}{Y \vdash y : b \sqsubseteq a' \to sa' \neq a}}{X, R, Y \vdash (r)xy : sa' \neq a}}{R, Y \vdash \lambda x(r)xy : \forall b'(a' \sqsubseteq b' \to sb' \neq sb) \to sa' \neq a}}{R, Y \vdash \lambda x(r)xy : \forall a'(\forall b'(a' \sqsubseteq b' \to sb' \neq sb) \to sa' \neq a)}}{R, Y \vdash \lambda x(r)xy : a \sqsubseteq sb}\,\mathrm{R}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{R, Y \vdash \lambda x(r)xy : a \sqsubseteq sb} \quad U \vdash u : a \sqsubseteq sb \to \bot}{R, Y, U \vdash (u)\lambda x(r)xy : \bot}}{R, U \vdash \lambda y(u)\lambda x(r)xy : \forall a'(b \sqsubseteq a' \to sa' \neq a) \to \bot}}{R, U \vdash \lambda y(u)\lambda x(r)xy : \forall b'(\forall a'(b' \sqsubseteq a' \to sa' \neq a) \to sb' \neq sb)}\,\mathrm{R}}{R, U \vdash \lambda y(u)\lambda x(r)xy : sb \sqsubseteq a}\,\mathrm{R} \quad V \vdash v : sb \sqsubseteq a \to \bot}{R, U, V \vdash (v)\lambda y(u)\lambda x(r)xy : \bot}}{R, U \vdash \lambda v(v)\lambda y(u)\lambda x(r)xy : (sb \sqsubseteq a \to \bot) \to \bot}}{R \vdash \lambda u\lambda v(v)\lambda y(u)\lambda x(r)xy : (a \sqsubseteq sb \to \bot), (sb \sqsubseteq a \to \bot) \to \bot}}{R \vdash \lambda u\lambda v(v)\lambda y(u)\lambda x(r)xy : F(sb)}}{\vdash \lambda r\lambda u\lambda v(v)\lambda y(u)\lambda x(r)xy : F(b) \to F(sb)}}{\vdash \lambda r\lambda u\lambda v(v)\lambda y(u)\lambda x(r)xy : \forall b(F(b) \to F(sb))}$$

Montrons enfin que $\lambda r\lambda u\lambda v(v)\lambda y(u)\lambda x(r)xy$ réalise $\top \to F(0)$. On prend les notations suivantes :

- R dénote le contexte $r : \top$,
- U dénote le contexte $u : a \sqsubseteq 0 \to \bot$,
- V dénote le contexte $v : 0 \sqsubseteq a \to \bot$.

La dérivation suivante donne le résultat.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{R, U \vdash \lambda y(u)\lambda x(r)xy : \top}{R, U \vdash \lambda y(u)\lambda x(r)xy : 0 \sqsubseteq a}\,\mathrm{R} \quad V \vdash v : 0 \sqsubseteq a \to \bot}{R, U, V \vdash (v)\lambda y(u)\lambda x(r)xy : \bot}}{R, U \vdash \lambda v(v)\lambda y(u)\lambda x(r)xy : (0 \sqsubseteq a \to \bot) \to \bot}}{R \vdash \lambda u\lambda v(v)\lambda y(u)\lambda x(r)xy : (a \sqsubseteq 0 \to \bot), (0 \sqsubseteq a \to \bot) \to \bot}}{R \vdash \lambda u\lambda v(v)\lambda y(u)\lambda x(r)xy : F(0)}}{\vdash \lambda r\lambda u\lambda v(v)\lambda y(u)\lambda x(r)xy : \top \to F(0)}$$

□

On considère maintenant le théorème 3.2.10. On se donne jusqu'à la fin de cette annexe les règles de remplacement permettant d'utiliser les égalités entre valeurs de vérité données dans la définition 3.2.1.

Théorème.
La formule $\forall x \forall x' \forall x'' \forall z (A_{\sqsubseteq} x', x'', x,\ A_{\sqsupseteq} x', x'', z \to z \sqsubseteq x)$ *est réalisée par la quasi-preuve suivante :*

$$(Y)\lambda r \lambda a \lambda b \lambda c \left((b)\lambda x(x)\lambda y((a)\dot{1})\lambda z(c)(r)zy\right) \lambda x(x)\lambda y((a)\dot{0})\lambda z(c)(r)zy.$$

Démonstration. On appelle $F(x)$ la formule

$$\forall x' \forall x'' \forall z (A_{\sqsubseteq} x', x'', x,\ A_{\sqsupseteq} x', x'', z \to z \sqsubseteq x).$$

Nous allons prouver que la quasi-preuve

$$\lambda r \lambda a \lambda b \lambda c \left((b)\lambda x(x)\lambda y((a)\dot{1})\lambda z(c)(r)zy\right) \lambda x(x)\lambda y((a)\dot{0})\lambda z(c)(r)zy$$

réalise les formules $\forall x(F(x) \to F(sx))$ et $\top \to F(0)$, ce qui donnera le résultat grâce au corollaire 2.2.19.

i) On considère pour commencer la formule $\forall x(F(x) \to F(sx))$. On pose
$H_1(x', x'', z) = \forall \bar{z}(\forall \bar{x}'(A_{\sqsupseteq} \bar{x}', x'', \bar{z} \to s\bar{x}' \neq x') \to s\bar{z} \neq z)$
et $H_0(x', x'', z) = \forall \bar{z}(\forall \bar{x}''(A_{\sqsupseteq} x', \bar{x}'', \bar{z} \to s\bar{x}'' \neq x'') \to s\bar{z} \neq z)$.
On prend de plus les notations suivantes :

- R dénote le contexte $r : F(x)$,
- A dénote le contexte $a : A_{\sqsubseteq} x', x'', sx$,
- B dénote le contexte $b : A_{\sqsupseteq} x', x'', z$,
- C dénote le contexte $c : \forall \bar{x}(\tilde{z} \sqsubseteq \bar{x} \to s\bar{x} \neq sx)$.

Nous allons déjà démontrer que le résultat se déduit des deux séquents
$R, A, C \vdash \lambda x(x)\lambda y((a)\dot{1})\lambda z(c)(r)zy : H_1(x', x'', z) \to s\tilde{z} \neq z$
et $R, A, C \vdash \lambda x(x)\lambda y((a)\dot{0})\lambda z(c)(r)zy : H_0(x', x'', z) \to s\tilde{z} \neq z$.
On notera ξ_1 et ξ_2 les termes apparaissant dans ces deux derniers séquents.
On a déjà la dérivation suivante :

$$\dfrac{\dfrac{B \vdash b : \forall X\{(H_1(x', x'', z) \to X), (H_0(x', x'', z) \to X) \to\ X\}}{B \vdash b : (H_1(x', x'', z) \to s\tilde{z} \neq z), (H_0(x', x'', z) \to s\tilde{z} \neq z) \to s\tilde{z} \neq z} \quad R, A, C \vdash \xi_1 : H_1(x', x'', z) \to s\tilde{z} \neq z}{B, R, A, C \vdash (b)\xi_1 : (H_0(x', x'', z) \to s\tilde{z} \neq z) \to s\tilde{z} \neq z}$$

Si l'on dispose du séquent $R, A, C \vdash \xi_2 : H_0(x', x'', z) \to s\tilde{z} \neq z$, on peut terminer comme suit :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{B,R,A,C \vdash (b)\xi_1\xi_2 : s\tilde{z} \neq z}{B,R,A \vdash \lambda c(b)\xi_1\xi_2 : \forall \bar{x}(\tilde{z} \sqsubseteq \bar{x} \to s\bar{x} \neq sx) \to s\tilde{z} \neq z}}{B,R,A \vdash \lambda c(b)\xi_1\xi_2 : \forall \tilde{z}(\forall \bar{x}(\tilde{z} \sqsubseteq \bar{x} \to s\bar{x} \neq sx) \to s\tilde{z} \neq z)}\,\mathrm{R}}{B,R,A \vdash \lambda c(b)\xi_1\xi_2 : z \sqsubseteq sx}}{R,A \vdash \lambda b\lambda c(b)\xi_1\xi_2 : A_{\sqsupseteq}x', x'', z \to z \sqsubseteq sx}}{R \vdash \lambda a\lambda b\lambda c(b)\xi_1\xi_2 : A_{\sqsubseteq}x', x'', sx,\ A_{\sqsupseteq}x', x'', z \to z \sqsubseteq sx}}{R \vdash \lambda a\lambda b\lambda c(b)\xi_1\xi_2 : F(sx)}}{\vdash \lambda r\lambda a\lambda b\lambda c(b)\xi_1\xi_2 : F(x) \to F(sx)}}{\vdash \lambda r\lambda a\lambda b\lambda c(b)\xi_1\xi_2 : \forall x(F(x) \to F(sx))}$$

Montrons maintenant que le séquent

$$R, A, C \vdash \lambda x(x)\lambda y((a)\dot{1})\lambda z(c)(r)zy : H_1(x', x'', z) \to s\tilde{z} \neq z$$

est dérivable. On reprend pour cela les notations précédentes, et l'on considère de plus que :

- X dénote le contexte $x : H_1(x', x'', z)$,
- Y dénote le contexte $y : A_{\sqsupseteq}\bar{x}', x'', \tilde{z}$,
- Z dénote le contexte $z : A_{\sqsubseteq}\bar{x}', x'', x$.

$$\dfrac{\dfrac{R \vdash r : \forall x'\forall x''\forall z(A_{\sqsubseteq}x', x'', x, A_{\sqsupseteq}x', x'', z \to z \sqsubseteq x)}{R \vdash r : A_{\sqsubseteq}\bar{x}', x'', x, A_{\sqsupseteq}\bar{x}', x'', \tilde{z} \to \tilde{z} \sqsubseteq x} \quad Z \vdash z : A_{\sqsubseteq}\bar{x}', x'', x}{R, Z \vdash (r)z : A_{\sqsupseteq}\bar{x}', x'', \tilde{z} \to \tilde{z} \sqsubseteq x} \quad Y \vdash y : A_{\sqsupseteq}\bar{x}', x'', \tilde{z}}{R, Z, Y \vdash (r)zy : \tilde{z} \sqsubseteq x}$$

La preuve se continue comme suit :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{R, Z, Y \vdash (r)zy : \tilde{z} \sqsubseteq x} \quad \dfrac{C \vdash c : \forall \bar{x}(\tilde{z} \sqsubseteq \bar{x} \to s\bar{x} \neq sx)}{C \vdash c : \tilde{z} \sqsubseteq x \to sx \neq sx}}{R, Z, Y, C \vdash (c)(r)zy : sx \neq sx}}{R, Y, C \vdash \lambda z(c)(r)zy : A_{\sqsubseteq}\bar{x}', x'', x \to sx \neq sx}}{R, Y, C \vdash \lambda z(c)(r)zy : \forall \bar{x}(A_{\sqsubseteq}\bar{x}', x'', \bar{x} \to s\bar{x} \neq sx)}\,\mathrm{R}$$

Partant du séquent $A \vdash a : A_{\sqsubseteq}x', x'', sx$, on dérive aisément le séquent $A \vdash (a)\dot{1} : \forall \bar{x}'(\forall \bar{x}(A_{\sqsubseteq}\bar{x}', x'', \bar{x} \to s\bar{x} \neq sx) \to s\bar{x}' \neq x')$ en utilisant le lemme 3.2.9. On peut alors conclure comme suit

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{R,Y,C,A\vdash (a)\dot{1}\lambda z(c)(r)zy : s\bar{x}'\neq x'}}{R,C,A\vdash \lambda y(a)\dot{1}\lambda z(c)(r)zy : A_{\sqsupseteq}\bar{x}',x'',\tilde{z}\to s\bar{x}'\neq x'}}{R,C,A\vdash \lambda y(a)\dot{1}\lambda z(c)(r)zy : \forall\bar{x}'(A_{\sqsupseteq}\bar{x}',x'',\tilde{z}\to s\bar{x}'\neq x')} \qquad X\vdash x : H_1(x',x'',z)}{R,C,A,X\vdash (x)\lambda y(a)\dot{1}\lambda z(c)(r)zy : s\tilde{z}\neq z}}{R,C,A\vdash \lambda x(x)\lambda y(a)\dot{1}\lambda z(c)(r)zy : H_1(x',x'',z)\to s\tilde{z}\neq z}$$

On démontre de manière analogue que le séquent

$$R,A,C\vdash \lambda x(x)\lambda y((a)\dot{0})\lambda z(c)(r)zy : H_0(x',x'',z)\to s\tilde{z}\neq z$$

est dérivable.

On reprend pour cela les notations R,A,C, et l'on considère de plus que :
- X dénote le contexte $x : H_0(x',x'',z)$.
- Y dénote le contexte $y : A_{\sqsupseteq}x',\bar{x}'',\tilde{z}$.
- Z dénote le contexte $z : A_{\sqsubseteq}x',\bar{x}'',x$.

On considère pour commencer la dérivation suivante :

$$\dfrac{\dfrac{R\vdash r : \forall x'\forall x''\forall z(A_{\sqsubseteq}x',x'',x,A_{\sqsupseteq}x',x'',z\to z\sqsubseteq x)}{R\vdash r : A_{\sqsubseteq}x',\bar{x}'',x,A_{\sqsupseteq}x',\bar{x}'',\tilde{z}\to\tilde{z}\sqsubseteq x} \qquad Z\vdash z : A_{\sqsubseteq}x',\bar{x}'',x}{R,Z\vdash (r)z : A_{\sqsupseteq}x',\bar{x}'',\tilde{z}\to\tilde{z}\sqsubseteq x} \qquad Y\vdash y : A_{\sqsupseteq}x',\bar{x}'',\tilde{z}$$
$$R,Z,Y\vdash (r)zy : \tilde{z}\sqsubseteq x$$

La preuve se continue comme suit

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{R,Z,Y\vdash (r)zy : \tilde{z}\sqsubseteq x} \qquad \dfrac{C\vdash c : \forall\bar{x}(\tilde{z}\sqsubseteq\bar{x}\to s\bar{x}\neq sx)}{C\vdash c : \tilde{z}\sqsubseteq x\to sx\neq sx}}{R,Z,Y,C\vdash (c)(r)zy : sx\neq sx}}{R,Y,C\vdash \lambda z(c)(r)zy : A_{\sqsubseteq}x',\bar{x}'',x\to sx\neq sx}}{R,Y,C\vdash \lambda z(c)(r)zy : \forall\bar{x}(A_{\sqsubseteq}x',\bar{x}'',\bar{x}\to s\bar{x}\neq sx)}\ \mathrm{R}$$

Le lemme 3.2.9 permet de montrer que le séquent
$A\vdash (a)\dot{0} : \forall\bar{x}''(\forall\bar{x}(A_{\sqsubseteq}x',\bar{x}'',\bar{x}\to s\bar{x}\neq sx)\to s\bar{x}''\neq x'')$ est dérivable, ce qui permet de conclure :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{R,Y,C,A\vdash (a)\dot{0}\lambda z(c)(r)zy : s\bar{x}''\neq x''}}{R,C,A\vdash \lambda y(a)\dot{0}\lambda z(c)(r)zy : A_{\sqsupseteq}x',\bar{x}'',\tilde{z}\to s\bar{x}''\neq x''}}{R,C,A\vdash \lambda y(a)\dot{0}\lambda z(c)(r)zy : \forall\bar{x}''(A_{\sqsupseteq}x',\bar{x}'',\tilde{z}\to s\bar{x}''\neq x'')} \qquad X\vdash x : A\vdash (a)\dot{0} : H_0(x',x'',z)}{R,C,A,X\vdash (x)\lambda y(a)\dot{0}\lambda z(c)(r)zy : s\tilde{z}\neq z}}{R,C,A\vdash \lambda x(x)\lambda y(a)\dot{0}\lambda z(c)(r)zy : H_0(x',x'',z)\to s\tilde{z}\neq z}$$

ii) Il ne nous reste donc plus qu'à prouver que la quasi-preuve

$$\lambda r\lambda a\lambda b\lambda c\left((b)\lambda x(x)\lambda y((a)\dot{1})\lambda z(c)(r)zy\right)\lambda x(x)\lambda y((a)\dot{0})\lambda z(c)(r)zy$$

réalise $\top \to F(0)$.
On pose les notations suivantes :

- R dénote le contexte $r : \top$,
- A dénote le contexte $a : A_{\sqsubseteq}x', x'', 0$,
- B dénote le contexte $b : A_{\sqsupseteq}x', x'', z$,
- Y dénote le contexte $A_{\sqsupseteq}x', \bar{x}'', \tilde{z}$,
- Z dénote le contexte $z : A_{\sqsubseteq}x', \bar{x}'', \bar{x}$,
- C dénote le contexte $c : \forall\bar{x}(\tilde{z} \sqsubseteq \bar{x} \to s\bar{x} \neq 0)$.

On reprend les notations H_1, ξ_1, H_0 et ξ_0 du cas précédent. Montrons déjà comment déduire le résultat des séquents $R, C, A \vdash \xi_0 : H_0(x', x'', z) \to s\tilde{z} \neq z$ et $R, C, A \vdash \xi_1 : H_1(x', x'', z) \to s\tilde{z} \neq z$.
La dérivation commence comme dans le cas précédent et se termine comme suit :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{B, R, A, C \vdash (b)\xi_1\xi_2 : s\tilde{z} \neq z}}{B, R, A \vdash \lambda c(b)\xi_1\xi_2 : \forall\bar{x}(\tilde{z} \sqsubseteq \bar{x} \to s\bar{x} \neq 0) \to s\tilde{z} \neq z}}{B, R, A \vdash \lambda c(b)\xi_1\xi_2 : \forall\tilde{z}(\forall\bar{x}(\tilde{z} \sqsubseteq \bar{x} \to s\bar{x} \neq 0) \to s\tilde{z} \neq z)}\,\mathrm{R}}{B, R, A \vdash \lambda c(b)\xi_1\xi_2 : z \sqsubseteq 0}}{R, A \vdash \lambda b\lambda c(b)\xi_1\xi_2 : A_{\sqsupseteq}x', x'', z \to z \sqsubseteq 0}}{R \vdash \lambda a\lambda b\lambda c(b)\xi_1\xi_2 : A_{\sqsubseteq}x', x'', 0,\ A_{\sqsupseteq}x', x'', z \to z \sqsubseteq 0}}{R \vdash \lambda a\lambda b\lambda c(b)\xi_1\xi_2 : F(0)}}{\vdash \lambda r\lambda a\lambda b\lambda c(b)\xi_1\xi_2 : \top \to F(0)}$$

Montrons maintenant comment dériver le séquent

$$R, C, A \vdash \xi_0 : H_0(x', x'', z) \to s\tilde{z} \neq z.$$

La dérivation commence comme suit

$$\dfrac{\dfrac{\dfrac{R, Z, Y, C \vdash (c)(r)zy : \top}{R, Z, Y, C \vdash (c)(r)zy : s\bar{x} \neq 0}}{R, Y, C \vdash \lambda z(c)(r)zy : A_{\sqsubseteq}x', \bar{x}'', \bar{x} \to s\bar{x} \neq 0}}{R, Y, C \vdash \lambda z(c)(r)zy : \forall\bar{x}(A_{\sqsubseteq}x', \bar{x}'', \bar{x} \to s\bar{x} \neq 0)}\,\mathrm{R}$$

Le lemme 3.2.9 permet alors de prouver que le séquent

$$A \vdash (a)\dot{0} : \forall\bar{x}(A_{\sqsubseteq}x', \bar{x}'', \bar{x} \to s\bar{x} \neq 0) \to s\bar{x}'' \neq x''$$

est dérivable. On peut alors écrire :

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{R,Y,C\vdash\lambda z(c)(r)zy:\forall\bar{x}(A_{\sqsubseteq}x',\bar{x}'',\bar{x}\to s\bar{x}\neq 0)}\quad A\vdash(a)\dot{0}:\forall\bar{x}(A_{\sqsubseteq}x',\bar{x}'',\bar{x}\to s\bar{x}\neq 0)\to s\bar{x}''\neq x''}{R,Y,C,A\vdash(a)\dot{0}\lambda z(c)(r)zy:s\bar{x}''\neq x''}}{R,C,A\vdash\lambda y(a)\dot{0}\lambda z(c)(r)zy:A_{\sqsupseteq}x',\bar{x}'',\tilde{z}\to s\bar{x}''\neq x''}}{R,C,A\vdash\lambda y(a)\dot{0}\lambda z(c)(r)zy:\forall\bar{x}''(A_{\sqsupseteq}x',\bar{x}'',\tilde{z}\to s\bar{x}''\neq x'')}$$

Considérant enfin le séquent

$$X\vdash x:\forall\tilde{z}(\forall\bar{x}''(A_{\sqsupseteq}x',\bar{x}'',\tilde{z}\to s\bar{x}''\neq x'')\to s\tilde{z}\neq z)$$

on en déduit que le séquent suivant est dérivable :

$$R,C,A\vdash\lambda x(x)\lambda y(a)\dot{0}\lambda z(c)(r)zy:H_0(x',x'',z)\to s\tilde{z}\neq z.$$

On démontrerait de même que le séquent

$$R,A,C\vdash\lambda x(x)\lambda y((a)\dot{1})\lambda z(c)(r)zy:H_1(x',x'',z)\to s\tilde{z}\neq z$$

est dérivable, ce qui donne le résultat. □

Donnons enfin les présentations en séquents des preuves des lemmes 3.2.12 et 3.2.13.

Lemme.
La formule $\forall x'\forall x''\forall x(A_{\sqsubseteq}x',x'',x\to A_{\sqsubseteq}x'',x',x)$ *est réalisée par la quasi-preuve suivante :*

$$(Y)\lambda r\lambda u\lambda v(u)\lambda a\lambda b\,((v)\lambda d(b)\lambda e(d)(r)e)\,\lambda d(a)\lambda e(d)(r)e.$$

Démonstration. On appelle $F(x)$ la formule

$$\forall x'\forall x''(A_{\sqsubseteq}x',x'',x\to A_{\sqsubseteq}x'',x',x).$$

Nous allons prouver que la quasi-preuve
$\lambda r\lambda u\lambda v(u)\lambda a\lambda b\,((v)\lambda d(b)\lambda e(d)(r)e)\,\lambda d(a)\lambda e(d)(r)e$ réalise les formules
$\forall x(F(x)\to F(sx))$ et $\top\to F(0)$, ce qui donnera le résultat grâce au corollaire 2.2.19.

i) On considère pour commencer la formule $\forall x(F(x) \to F(sx))$. Soit X une variable propositionnelle. On pose
$H_1(x'', x', x) = \forall \bar{x}''(\forall \bar{x}(A_\sqsubseteq \bar{x}'', x', \bar{x} \to s\bar{x} \neq sx) \to s\bar{x}'' \neq x'')$ et
$H_0(x'', x', x) = \forall \bar{x}'(\forall \bar{x}(A_\sqsubseteq x'', \bar{x}', \bar{x} \to s\bar{x} \neq sx) \to s\bar{x}' \neq x')$.
On prend encore les notations suivantes :

- R dénote le contexte $r : F(x)$,
- U dénote le contexte $u : A_\sqsubseteq x', x'', sx$,
- A dénote le contexte $a : H_1(x', x'', x)$,
- B dénote le contexte $b : H_0(x', x'', x)$,
- V dénote le contexte $v : H_1(x'', x', x), H_0(x'', x', x) \to X$.

Montrons déjà que l'on peut prouver le résultat à partir des séquents
$R, B \vdash \lambda d(b)\lambda e(d)(r)e : H_1(x'', x', x)$ et
$R, A \vdash \lambda d(a)\lambda e(d)(r)e : H_0(x'', x', x)$.
On note ξ_1 et ξ_2 les termes apparaissant dans ces deux derniers séquents. On considère déjà la dérivation suivante :

$$\dfrac{\dfrac{V \vdash v : H_1(x'', x', x), H_0(x'', x', x) \to X \qquad R, B \vdash \xi_1 : H_1(x'', x', x)}{V, R, B \vdash (v)\xi_1 : H_0(x'', x', x) \to X} \qquad R, A \vdash \xi_2 : H_0(x'', x', x)}{V, R, B, A \vdash ((v)\xi_1)\,\xi_2 : X}$$

On déduit ensuite des hypothèses faites sur a, b et u que

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{U, V, R \vdash (u)\lambda a\lambda b\,((v)\xi_1)\,\xi_2 : X}}{R, U \vdash \lambda v(u)\lambda a\lambda b\,((v)\xi_1)\,\xi_2 : (: H_1(x'', x', x), H_0(x'', x', x) \to X) \to X}}{R, U \vdash \lambda v(u)\lambda a\lambda b\,((v)\xi_1)\,\xi_2 : A_\sqsubseteq x'', x', sx}}{R \vdash \lambda u\lambda v(u)\lambda a\lambda b\,((v)\xi_1)\,\xi_2 : A_\sqsubseteq x', x'', sx \to A_\sqsubseteq x'', x', sx}}{R \vdash \lambda u\lambda v(u)\lambda a\lambda b\,((v)\xi_1)\,\xi_2 : F(sx)}}{\vdash \lambda r\lambda u\lambda v\lambda a\lambda b\,((u)\xi_1)\,\xi_2 : F(x) \to F(sx)}}{\vdash \lambda r\lambda u\lambda a\lambda b\,((u)\xi_1)\,\xi_2 : \forall x(F(x) \to F(sx))}$$

Prouvons maintenant que le séquent

$$R, B \vdash \lambda d(b)\lambda e(d)(r)e : H_1(x'', x', x)$$

est dérivable. On note D le contexte $d : \forall \bar{x}(A_\sqsubseteq \tilde{x}'', x', \bar{x} \to s\bar{x} \neq sx)$ et E le contexte $e : A_\sqsubseteq x', \tilde{x}'', x$.

$$\dfrac{\dfrac{\dfrac{\dfrac{R \vdash r : A_\sqsubseteq x', \tilde{x}'', x \to A_\sqsubseteq \tilde{x}'', x', x \qquad E \vdash e : A_\sqsubseteq x', \tilde{x}'', x}{R, E \vdash (r)e : A_\sqsubseteq \tilde{x}'', x', x} \qquad \dfrac{D \vdash d : \forall \bar{x}(A_\sqsubseteq \tilde{x}'', x', \bar{x} \to s\bar{x} \neq sx)}{D \vdash d : A_\sqsubseteq \tilde{x}'', x', x \to sx \neq sx}}{R, E, D \vdash (d)(r)e : sx \neq sx}}{R, D \vdash \lambda e(d)(r)e : A_\sqsubseteq x', \tilde{x}'', x \to sx \neq sx}}{R, D \vdash \lambda e(d)(r)e : \forall \bar{x}(A_\sqsubseteq x', \tilde{x}'', \bar{x} \to s\bar{x} \neq sx)}\ \mathrm{R}$$

La preuve continue comme suit.

$$\dfrac{\dfrac{\dfrac{\vdots}{R, D \vdash \lambda e(d)(r)e : \forall \bar{x}(A_{\sqsubseteq} x', \tilde{x}'', \bar{x} \to s\bar{x} \neq sx)} \quad \dfrac{B \vdash b : H_0(x', x'', x)}{B \vdash b : \forall \bar{x}(A_{\sqsubseteq} x', \bar{x}'', \bar{x} \to s\bar{x} \neq sx) \to s\bar{x}'' \neq x''}}{R, D, B \vdash (b)\lambda e(d)(r)e : s\tilde{x}'' \neq x''}}{R, B \vdash \lambda d(b)\lambda e(d)(r)e : \forall \bar{x}(A_{\sqsubseteq} \tilde{x}'', x', \bar{x} \to s\bar{x} \neq sx) \to s\tilde{x}'' \neq x''}}{R, B \vdash \lambda d(b)\lambda e(d)(r)e : H_1(x'', x', x)}$$

On prouverait de même que le séquent $R, A \vdash \lambda d(a)\lambda e(d)(r)e : H_2(x'', x', x)$ est dérivable.

ii) On considère maintenant la formule $\top \to F(0)$. On pose
$H_1(x'', x') = \forall \bar{x}''(\forall \bar{x}(A_{\sqsubseteq} \bar{x}'', x', \bar{x} \to s\bar{x} \neq 0) \to s\bar{x}'' \neq x'')$ et
$H_0(x'', x') = \forall \bar{x}'(\forall \bar{x}(A_{\sqsubseteq} x'', \bar{x}', \bar{x} \to s\bar{x} \neq 0) \to s\bar{x}' \neq x')$. On note encore R le contexte $r : \top$, A le contexte $a : H_1(x', x'')$, et B le contexte $b : H_2(x', x'')$. En raisonnant comme dans le cas précédent, on peut montrer que le résultat découle des deux séquents
$R, B \vdash \lambda d(b)\lambda e(d)(r)e : H_1(x'', x')$
et $R, A \vdash \lambda d(a)\lambda e(d)(r)e : H_0(x'', x')$.

Prouvons le premier, la preuve du second étant analogue. On note D le contexte $d : \forall \bar{x}(A_{\sqsubseteq} \tilde{x}'', x', \bar{x} \to s\bar{x} \neq 0)$ et E le contexte $e : A_{\sqsubseteq} x', \tilde{x}'', \bar{x}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{R, E, D \vdash (d)(r)e : \top}{R, E, D \vdash (d)(r)e : s\bar{x} \neq 0}\,\mathrm{R}}{R, D \vdash \lambda e(d)(r)e : A_{\sqsubseteq} x', \tilde{x}'', \bar{x} \to s\bar{x} \neq 0}}{R, D \vdash \lambda e(d)(r)e : \forall \bar{x}(A_{\sqsubseteq} x', \tilde{x}'', \bar{x} \to s\bar{x} \neq 0)} \quad \dfrac{B \vdash b : H_0(x', x'')}{B \vdash b : \forall \bar{x}''(\forall \bar{x}(A_{\sqsubseteq} x', \bar{x}'', \bar{x} \to s\bar{x} \neq 0) \to s\bar{x}'' \neq x'')}}{R, D, B \vdash (b)\lambda e(d)(r)e : s\tilde{x}'' \neq x''}}{R, B \vdash \lambda d(b)\lambda e(d)(r)e : \forall \bar{x}(A_{\sqsubseteq} \tilde{x}'', x', \bar{x} \to s\bar{x} \neq 0) \to s\tilde{x}'' \neq x''}}{R, B \vdash \lambda d(b)\lambda e(d)(r)e : H_1(x'', x')}$$

Ce qui achève la démonstration. □

Lemme.
La formule $\forall x'\forall x''\forall x(A_{\sqsupseteq}x', x'', x \to A_{\sqsupseteq}x'', x', x)$ *est réalisée par la quasi-preuve suivante :*

$$(Y)\lambda r\lambda u\lambda a\lambda b\,((u)\lambda c(b)\lambda d(c)\lambda e(d)(r)e)\,\lambda c(a)\lambda d(c)\lambda e(d)(r)e.$$

Démonstration. On note $F(x)$ la formule

$$\forall x'\forall x''(A_{\sqsupseteq}x', x'', x \to A_{\sqsupseteq}x'', x', x).$$

Nous allons prouver que la quasi-preuve
$\lambda r\lambda u\lambda a\lambda b\,((u)\lambda c(b)\lambda d(c)\lambda e(d)(r)e)\,\lambda c(a)\lambda d(c)\lambda e(d)(r)e$ réalise les formules $\forall x(F(x) \to F(sx))$ et $\top \to F(0)$, ce qui donnera le résultat grâce au corollaire 2.2.19.

i) On considère pour commencer la formule $\forall x(F(x) \to F(sx))$. Soit X une variable propositionnelle. On pose
$H_1(x', x'', x) = \forall\bar{x}(\forall\bar{x}'(A_{\sqsupseteq}\bar{x}', x'', \bar{x} \to s\bar{x}' \neq x') \to s\bar{x} \neq sx)$ et
$H_0(x', x'', x) = \forall\bar{x}(\forall\bar{x}''(A_{\sqsupseteq}x', \bar{x}'', \bar{x} \to s\bar{x}'' \neq x'') \to s\bar{x} \neq sx)$.
On prend encore les notations suivantes :

- R dénote le contexte $r : F(x)$,
- U dénote le contexte $u : A_{\sqsupseteq}x', x'', sx$,
- A dénote le contexte $a : H_1(x'', x', x) \to X$,
- B note le contexte $b : H_0(x'', x', x) \to X$.

Montrons déjà que l'on peut prouver le résultat à partir des séquents
$R, B \vdash \lambda c(b)\lambda d(c)\lambda e(d)(r)e : H_1(x', x'', x) \to X$
et $R, A \vdash \lambda c(a)\lambda d(c)\lambda e(d)(r)e : H_0(x', x'', x) \to X$.
On note ξ_1 et ξ_2 les termes apparaissant dans ces deux derniers séquents.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{U \vdash u : A_{\sqsupseteq}x', x'', sx}{U \vdash u : H_1(x', x'', x), H_0(x', x'', x) \to X}\,\mathrm{R} \quad R, B \vdash \xi_1 : H_1(x', x'', x)}{U, R, B \vdash (u)\xi_1 : H_0(x', x'', x) \to X} \quad R, A \vdash \xi_2 : H_0(x', x'', x)}{U, R, B, A \vdash ((u)\xi_1)\,\xi_2 : X}}{U, R, A \vdash \lambda b\,((u)\xi_1)\,\xi_2 : H_0(x'', x', x) \to X}}{U, R \vdash \lambda a\lambda b\,((u)\xi_1)\,\xi_2 : H_1(x'', x', x), H_0(x'', x', x) \to X}}{U, R \vdash \lambda a\lambda b\,((u)\xi_1)\,\xi_2 : A_{\sqsupseteq}x'', x', sx}}{R \vdash \lambda u\lambda a\lambda b\,((u)\xi_1)\,\xi_2 : A_{\sqsupseteq}x', x'', sx \to A_{\sqsupseteq}x'', x', sx}}{R \vdash \lambda u\lambda a\lambda b\,((u)\xi_1)\,\xi_2 : F(sx)}}{\vdash \lambda r\lambda u\lambda a\lambda b\,((u)\xi_1)\,\xi_2 : F(x) \to F(sx)}}{\vdash \lambda r\lambda u\lambda a\lambda b\,((u)\xi_1)\,\xi_2 : \forall x(F(x) \to F(sx))}$$

Prouvons maintenant que le séquent
$R, B \vdash \lambda c(b)\lambda d(c)\lambda e(d)(r)e : H_1(x', x'', x) \to X$ est dérivable. On reprend les notations ci-dessus, et l'on note encore C le contexte $c : H_1(x', x'', x)$, D le contexte $d : \forall\tilde{x}'(A_{\sqsupseteq}x'', \tilde{x}', x \to s\tilde{x}' \neq x')$ et E le contexte $e : A_{\sqsupseteq}\tilde{x}', x'', x$.

$$\dfrac{\dfrac{\dfrac{R \vdash r : F(x)}{R \vdash r : A_{\sqsupseteq}\tilde{x}', x'', x \to A_{\sqsupseteq}x'', \tilde{x}', x} \quad E \vdash e : A_{\sqsupseteq}\tilde{x}', x'', x}{R, E \vdash (r)e : A_{\sqsupseteq}x'', \tilde{x}', x} \quad \dfrac{D, \vdash d : \forall \bar{x}'(A_{\sqsupseteq}x'', \bar{x}', x \to s\bar{x}' \neq x')}{D, \vdash d : \forall A_{\sqsupseteq}x'', \tilde{x}', x \to s\tilde{x}' \neq x'}}{R, E, D \vdash (d)(r)e : s\tilde{x}' \neq x'}}{\dfrac{R, D \vdash \lambda e(d)(r)e : A_{\sqsupseteq}\tilde{x}', x'', x \to s\tilde{x}' \neq x'}{R, D \vdash \lambda e(d)(r)e : \forall x'(A_{\sqsupseteq}\tilde{x}', x'', x \to s\tilde{x}' \neq x')}}$$

La preuve se continue alors comme suit.

$$\dfrac{\dfrac{\dfrac{\vdots}{R, D \vdash \lambda e(d)(r)e : \forall x'(A_{\sqsupseteq}\tilde{x}', x'', x \to s\tilde{x}' \neq x')} \quad \dfrac{C \vdash c : H_1(x', x'', x)}{C \vdash c : \forall \bar{x}'(A_{\sqsupseteq}\bar{x}', x'', x \to s\bar{x}' \neq x') \to sx \neq sx}}{R, D, C \vdash (c)\lambda e(d)(r)e : sx \neq sx}}{\dfrac{R, C \vdash \lambda d(c)\lambda e(d)(r)e : \forall \bar{x}'(A_{\sqsupseteq}x'', \bar{x}', x \to s\bar{x}' \neq x') \to sx \neq sx}{R, C \vdash \lambda d(c)\lambda e(d)(r)e : H_0(x'', x', x)}\,\mathrm{R}}$$

On peut alors conclure :

$$\dfrac{\dfrac{\dfrac{\vdots}{R, C \vdash \lambda d(c)\lambda e(d)(r)e : H_0(x'', x', x)} \quad b : H_0(x'', x', x) \to X}{R, C, B \vdash (b)\lambda d(c)\lambda e(d)(r)e : X}}{R, B \vdash \lambda c(b)\lambda d(c)\lambda e(d)(r)e : H_1(x', x'', x) \to X}$$

Il reste enfin à démontrer que le séquent
$R, A \vdash \lambda c(a)\lambda d(c)\lambda e(d)(r)e : H_0(x', x'', x) \to X$ est dérivable, ce qui se fait de manière analogue.

ii) Montrons maintenant que $\top \to F(0)$ est réalisée par le terme considéré. On pose $H_1(x', x'') = \forall \bar{x}(\forall \bar{x}'(A_{\sqsupseteq}\bar{x}', x'', \bar{x} \to s\bar{x}' \neq x') \to s\bar{x} \neq 0)$ et
$H_0(x', x'') = \forall \bar{x}(\forall \bar{x}''(A_{\sqsupseteq}x', \bar{x}'', \bar{x} \to s\bar{x}'' \neq x'') \to s\bar{x} \neq 0)$. On prend encore les notations suivantes :

- R dénote le contexte $r : \top$,
- U dénote le contexte $u : A_{\sqsupseteq}x', x'', 0$,
- A dénote le contexte $a : H_1(x'', x') \to X$,
- B dénote le contexte $b : H_0(x'', x') \to X$.

Le résultat se démontre à partir des séquents
$R, B \vdash \lambda c(b)\lambda d(c)\lambda e(d)(r)e : H_1(x', x'') \to X$
et $R, A \vdash \lambda c(a)\lambda d(c)\lambda e(d)(r)e : H_0(x', x'') \to X$, la preuve étant la même que celle faite dans le cas précédent.
Prouvons encore une fois le premier de ces séquents, la preuve du second étant analogue.
On note C le contexte $c : H_1(x', x'')$, D le contexte $d : \forall \bar{x}'(A_{\sqsupseteq}x'', \bar{x}', \tilde{x} \to s\bar{x}' \neq x')$

$$\dfrac{\dfrac{\dfrac{R, D, C \vdash (c)\lambda e(d)(r)e : \top}{R, D, C \vdash (c)\lambda e(d)(r)e : s\bar{x} \neq 0}\,\mathrm{R}}{R, C \vdash \lambda d(c)\lambda e(d)(r)e : \forall \bar{x}'(A_{\sqsupseteq} x'', \bar{x}', \tilde{x} \to s\bar{x}' \neq x') \to s\tilde{x} \neq 0}}{R, C \vdash \lambda d(c)\lambda e(d)(r)e : H_0(x'', x')}$$

On peut ensuite conclure

$$\dfrac{\dfrac{\dfrac{\vdots}{R, C \vdash \lambda d(c)\lambda e(d)(r)e : H_0(x'', x')} \qquad B \vdash b : H_0(x'', x') \to X}{R, C, B \vdash (b)\lambda d(c)\lambda e(d)(r)e : X}}{R, B \vdash \lambda c(b)\lambda d(c)\lambda e(d)(r)e : H_1(x', x'') \to X}$$

Ce qui achève la démonstration. □

Bibliographie

[1] A. Arnold et D. Niwinski : *Rudiments of μ-calculus*. North-Holland, 2001.

[2] Henk Barendregt : *The Lambda Calculus. Its syntax and semantics*. Elsevier, 1984.

[3] Emmanuel Beffara : *Logique, réalisabilité et concurrence*. Thèse de doctorat, Université Paris Diderot-Paris 7, 2005.

[4] Felice Cardone et J. Roger Hindley : The history of lambda-calculus and combinatory logic. *Handbook of the History of Logic*, 5, 2006.

[5] René Cori et Daniel Lascar : *Logique mathématique*. Masson, 1993.

[6] Harvey Friedman : The consistency of classical set theory relative to a set theory with intuitionistic logic. *J. Symbolic Logic*, 38:315–319, 1973.

[7] Jean-Yves Girard : A new constructive logic : classical logic. *Math. Struct. Comput. Sci.*, 3:255–296, 1991.

[8] Timothy G. Griffin : A formulæ-as-type notion of control. *In Conf. Record 17th Annual ACM Symp. on Principles of Programming Languages, POPL'90*, 1990.

[9] Mauricio Guillermo : *Realizability games in arithmetical formulae*. Thèse de doctorat, Université Paris Diderot-Paris 7, 2008.

[10] Philippe Hesse : Herbrand's theorem in classical realizability. Submitted for publication, available at http://hal.archives-ouvertes.fr.

[11] Thomas J. Jech : *Set theory*. Academic Press, 1978.

[12] Jerome H. Keisler et Chen Chung Chang : *Model theory*. North Holland, 1990.

[13] Georg Kreisel : On the interpretation of non-finitists proofs part I. *J. Symbolic Logic*, 16:241–267, 1951.

[14] Georg Kreisel : On the interpretation of non-finitists proofs part II. *J. Symbolic Logic*, 17:43–58, 1952.

[15] Georg Kreisel : Mathematical significance of consistency proofs. *J. Symbolic Logic*, 23:155–182, 1958.

[16] Jean-Louis Krivine : The Curry-Howard correspondence in classical analysis and set theory. Diapositives d'un exposé à WoLLIC'05. Disponibles à http://www.pps.jussieu.fr/~krivine/.

[17] Jean-Louis Krivine : Realizability : a machine for analysis and set theory. Diapositives d'un exposé à Geocal'06. Disponibles à http://cel.archives-ouvertes.fr/cel-00154509/fr/.

[18] Jean-Louis Krivine : Realizability in classical logic. A paraître dans *Panoramas et synthèses*, SMF. Disponible à http://hal.archives-ouvertes.fr/hal-00154500/fr/.

[19] Jean-Louis Krivine : *Lambda-calculus, types and models*. Ellis Horwood, 1993.

[20] Jean-Louis Krivine : Classical logic, storage operators and second order lambda-calculus. *Pure and Appl. Log.*, 68:63–78, 1994.

[21] Jean-Louis Krivine : A general storage theorem for integers in call by name lambda-calculus. *Th. Comp. Sc.*, 129:79–94, 1994.

[22] Jean-Louis Krivine : Une preuve formelle et intuitionniste du théorème de complétude de la logique classique. *Bull. Symbolic Logic*, 308:405–421, 1996.

[23] Jean-Louis Krivine : *Théorie des ensembles*. Cassini, 1998.

[24] Jean-Louis Krivine : Typed lambda-calculus in classical Zermelo-Frænkel set theory. *Arch. Math. Log.*, 40:189–205, 2001.

[25] Jean-Louis Krivine : Dependent choice, 'quote' and the clock. *Th. Comp. Sc.*, 308:259–276, 2003.

[26] Jean-Louis Krivine : A call by name lambda-calculus machine. *Higher Order and Symbolic Computation*, 20:199–207, 2007.

[27] Jean-Louis Krivine et Vincent Danos : Disjunctive tautologies and synchronisation schemes. *In CSL*, volume 1862 de *LNCS*, pages 292–301, 2000.

[28] Jean-Louis Krivine et Yves Legrandgérard : Valid formulas, games and network protocols. A paraitre, disponible à http://hal.archives-ouvertes.fr/hal-00166880/fr/.

[29] Kenneth Kunen : *Set theory : an introduction to independence proofs*. North-Holland, 1980.

[30] Maria Manzano : *Extensions of first order logic*. Cambridge tracts in theoretical computer science, 1996.

[31] Michel Parigot : $\lambda\mu$-calculus : an algorithmic interpretation of classical natural deduction. *In Logic for Programming and Automated Reasoning*, volume 624 de *LNCS*, pages 190–201, 1992.

[32] Jon Postel : Transmission Control Protocol specification. *RFC 793*, 1981.

[33] Andrew S. Tanenbaum : *Computer networks*. Prentice Hall, 1996.

[34] Alfred Tarski : A lattice-theoretical fixpoint theorem and its applications. *Pacific J. Math.*, 5:285–309, 1955.

Index

Résumé

Cette thèse étudie différents aspects de la réalisabilité classique due à Jean-Louis Krivine. Celle-ci permet de mettre en œuvre l'isomorphisme de Curry-Howard : on peut ainsi associer un programme à chaque démonstration mathématique, et considérer chaque théorème comme une spécification. Dans un premier temps, on rappelle le formalisme de la réalisabilité classique ainsi que certains de ses résultats fondamentaux. On s'attache ensuite à l'analyse des contenus opérationnels obtenus suivant deux méthodes différentes d'étude des entiers des modèles de la réalisabilité. Dans un second temps, on rappelle la notion de jeu qui peut être associée à chaque formule du premier ordre dans ce cadre. Ces jeux permettent d'établir une correspondance entre les formules valides du calcul des prédicats et les protocoles de la couche transport des réseaux, que l'on peut spécifier de manière claire et précise par ce biais. La dernière partie est consacrée à l'étude de l'axiome du choix dépendant. On montre que la méthode développée pour le réaliser s'adapte à une expression simple de celui-ci au niveau des individus d'un modèle. On utilise enfin l'instruction associée pour réaliser un cas particulier du théorème de Herbrand. Le terme obtenu effectue une opération très générale, qui peut être interprétée dans le cadre des protocoles réseaux.

Abstract

This thesis studies different aspects of Jean-Louis Krivine's classical realizability. This principle allows one to carry out the Curry-Howard isomorphism : a program can be associated with each mathematical proof, and each theorem can be considered as a specification. In the first part, we recall the formalism of classical realizability and some of its fundamental results. We then analyze the operational contents obtained following two different methods of reasoning on the integers of a realizability model. Secondly, we recall the notion of game that can be associated with any first order formula in the framework of classical realizability. These games permit the establishement of a correspondence between valid formulas of predicate calculus and network protocols ; the latter ones can thus be specified in a clear and precise way. The last part is devoted to the study of the axiom of dependent choice. We show that the method used to realize it can be adapted to a simple expression of choice on the individuals of a model. Eventually, we use the associated instruction to realize a particular case of Herbrand's theorem. The term obtained effects a very general operation that can be interpreted in the framework of network protocols.